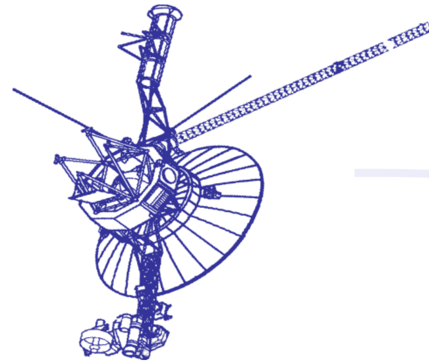
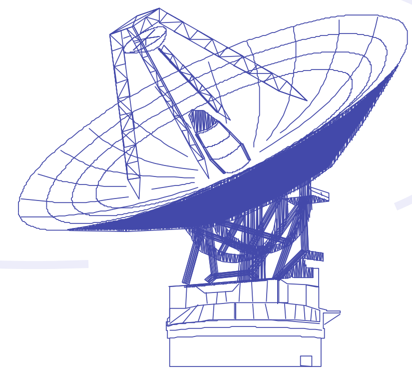
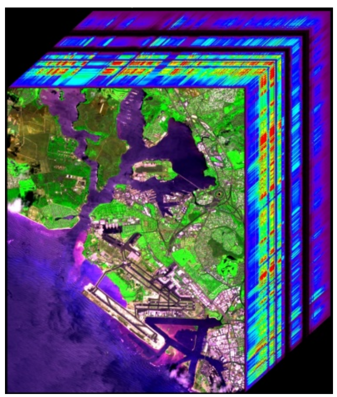
Real-time Airborne Demonstration of Fast Lossless Hyperspectral Data Compression System for AVIRIS-NG and PRISM







Didier Keymeulen, Huy Luong, Nazeeh Aranki, Charles Sarture, Michael Eastwood, Ian Mccubbin, Alan Mazer, Matt Klimesh, Robert Green, David Dolman (3), Alireza Bakhshi (2)

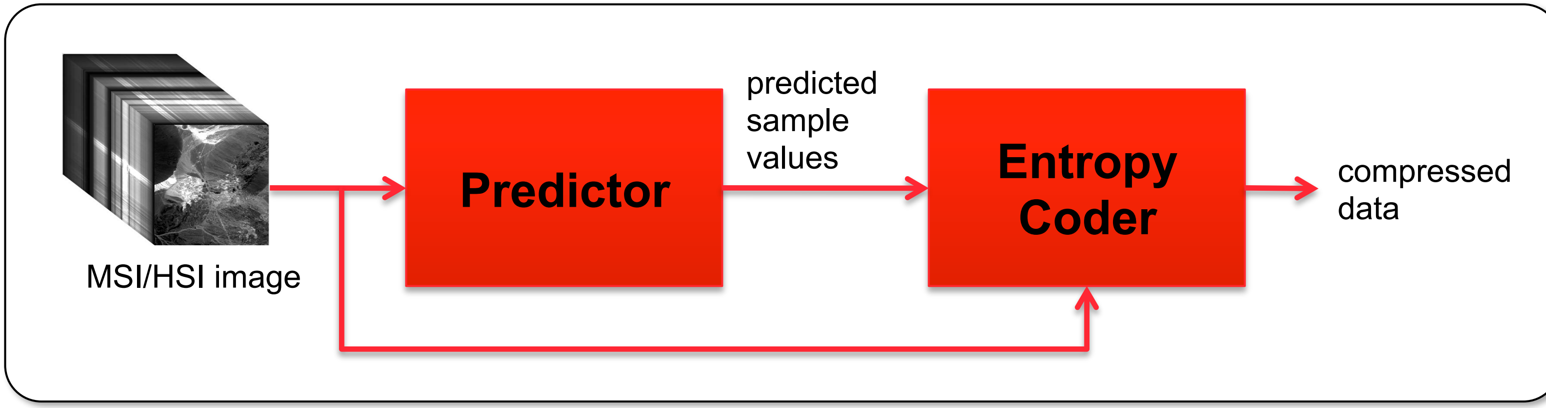
Jet Propulsion Laboratory
California Institute of Technology
(2) B&A Engineering Inc.
(3) Alpha Data Inc.

Outline

- Overview of Fast Lossless (FL) Hyperspectral Data Compression Algorithm
- Fast Lossless FPGA Implementation
- Airborne Demonstrations

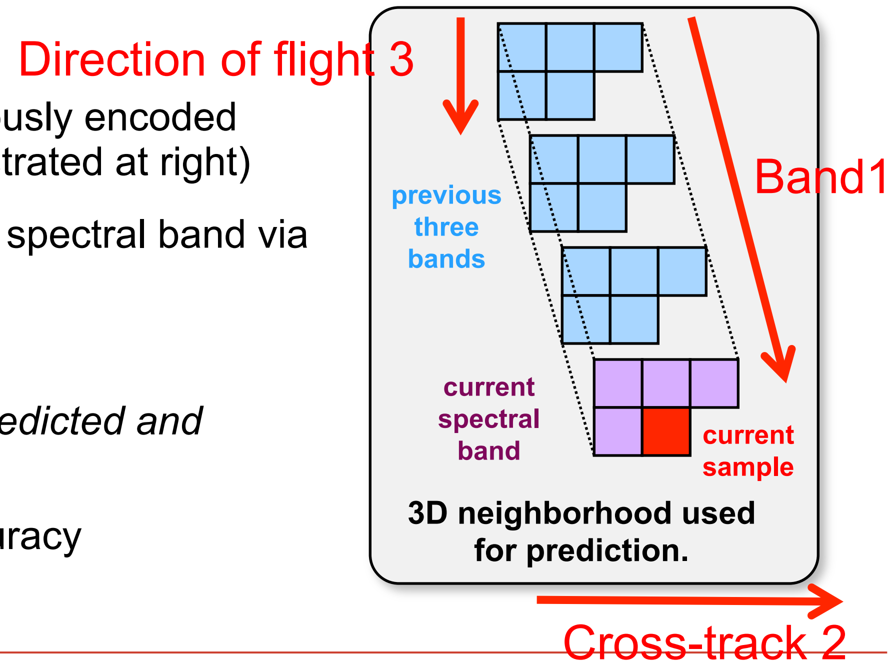
Fast Lossless (FL) MSI/HSI Compressor

FL Compressor Overview



Approach: Predictive compression, encoding samples one-at-a-time

- **Predictor**
  - Computes predicted sample value from previously encoded nearby samples (prediction neighborhood illustrated at right)
  - Adaptively adjusts prediction weights for each spectral band via adaptive linear prediction
- **Entropy Coder**
  - Losslessly encodes the *difference between predicted and actual sample values*
  - Adaptively adjusts to changing prediction accuracy



3D neighborhood used for prediction.

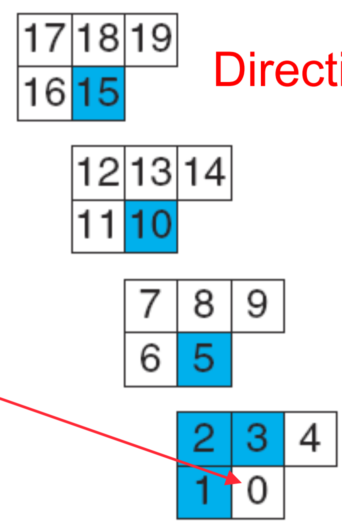
Compression Algorithm: Estimation

- **Purpose:** Estimate a desired signal d_t from an input vector $\vec{u}_t$ using a linear estimator that is adaptively updated from previous results
- **Compression of Estimate Error :**
  - Form estimate: $\hat{d}_t = \vec{w}_t^T \vec{u}_t$
  - Calculate estimation error: $e_t = \hat{d}_t - d_t$

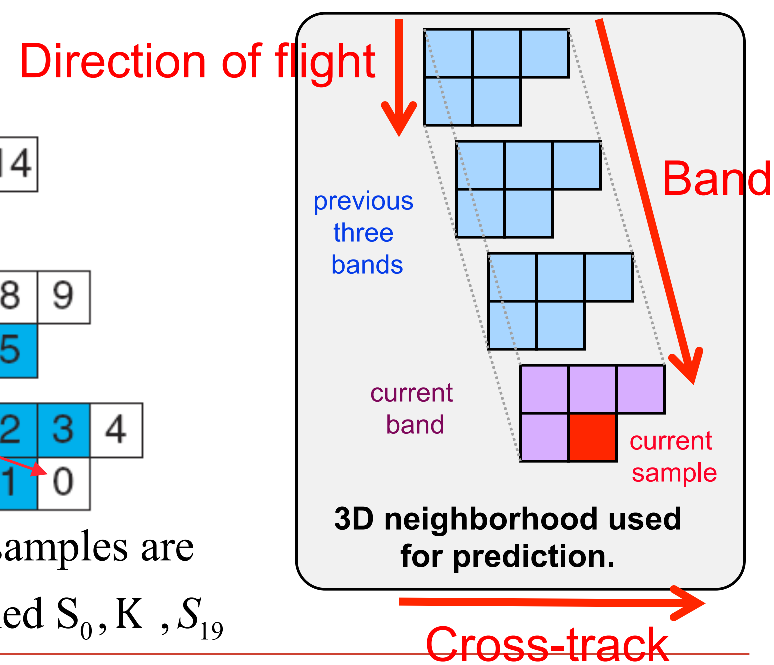
    e_t is encoded in the compressed bitstream
  - Update filter weights using the sign algorithm: $\vec{w}_{t+1} = \vec{w}_t - \mu \vec{u}_t \operatorname{sgn}(e_t)$

    where μ is the “adaptation step size” parameter
- **Naive approach:** use local neighborhood to construct $\vec{u}_t$ around $d_t = s_0$

with $d_t = s_0$ and $\vec{u}_t = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_5 \\ s_{10} \\ s_{15} \end{bmatrix}$



The samples are labelled $S_0, \ldots, S_{19}$



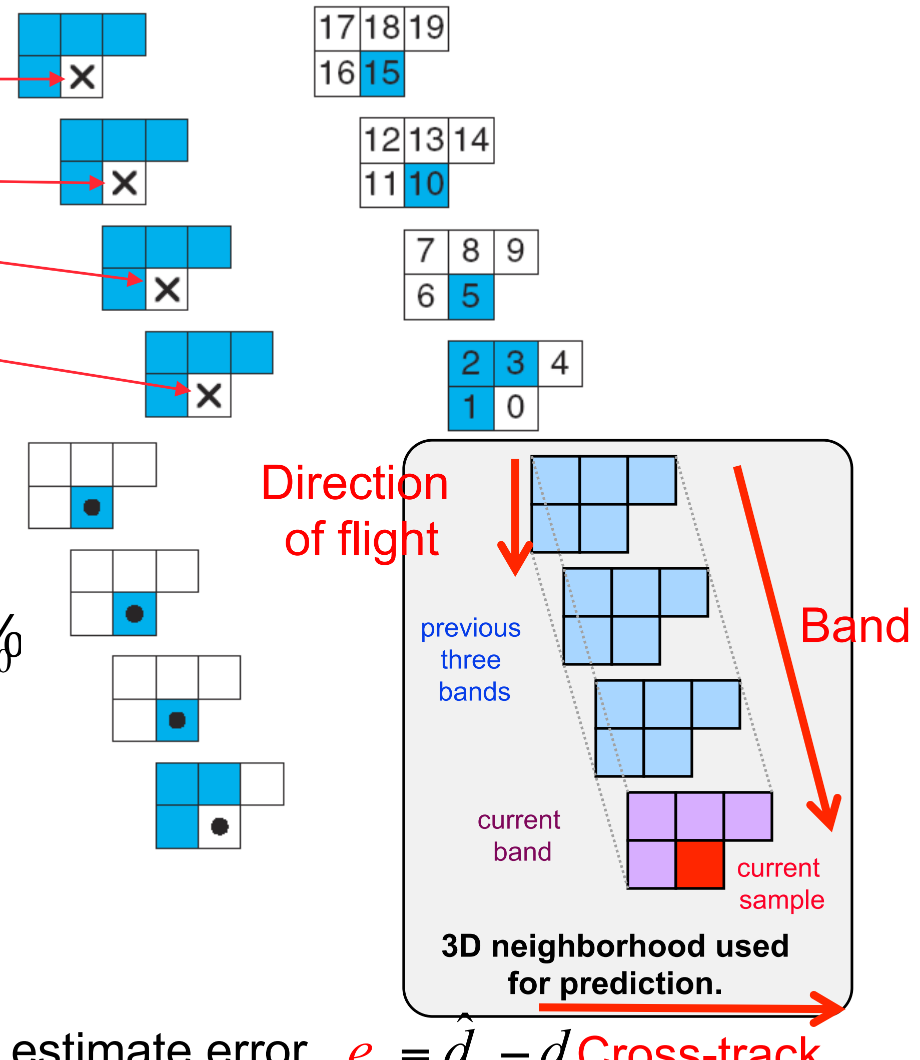
3D neighborhood used for prediction.

But performs poorly ….

Compression Algorithm: Local Mean Subtraction

- **Our solution**: compute simple preliminary estimates $\tilde{s}_i$ in each band at the spatial location of the sample being predicted, and subtract from the input samples.

$$\tilde{s}_{15} = (s_{16} + s_{17} + s_{18} + s_{19})/4$$
$$\tilde{s}_{10} = (s_{11} + s_{12} + s_{13} + s_{14})/4$$
$$\tilde{s}_{5} = (s_{6} + s_{7} + s_{8} + s_{9})/4$$
$$\tilde{s}_{0} = (s_{1} + s_{2} + s_{3} + s_{4})/4$$

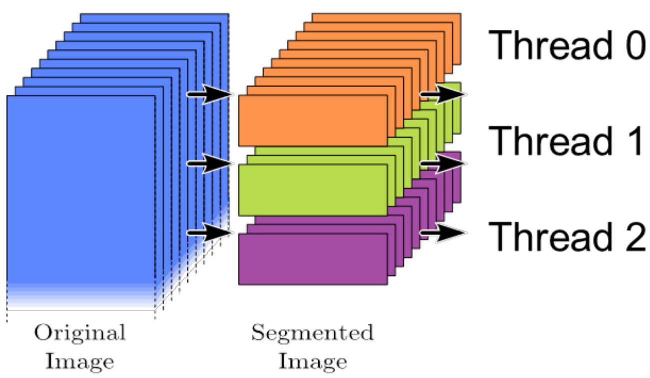


Use $\vec{u}_t = \begin{bmatrix} s_1 - \tilde{s}_0 \\ s_2 - \tilde{s}_0 \\ s_3 - \tilde{s}_0 \\ s_5 - \tilde{s}_5 \\ s_{10} - \tilde{s}_{10} \\ s_{15} - \tilde{s}_{15} \end{bmatrix}$ and $d_t = s_0 - \tilde{s}_0$

to compute the estimate $\hat{d}_t = \vec{w}_t^T \vec{u}_t$ and the estimate error $e_t = \hat{d}_t - d_t$

Compression Algorithm: Implementation

- **Sign algorithm** is used for weight adaptation
- **Estimation error** is encoded using Golomb power-of-2 codes
- **Dataset is divided into parts (32 lines each)**, which are compressed independently. This provides some error containment.
- **Each spectral band has its own prediction weights**, maintained independently of the prediction weights for other spectral bands



Compression Algorithm: Other Methods

Compare our **“Fast lossless”** compression algorithm with:

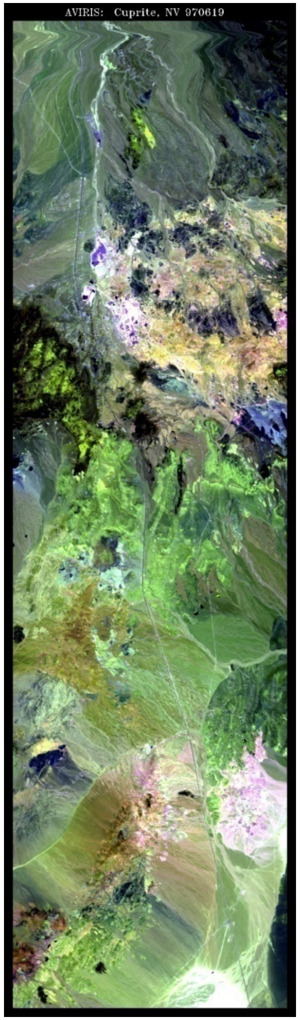
- ***ICER-3D***: *a 3-D-wavelet-based compressor which is the state-of-the-art (ICER-2D is used on both spirit and opportunity MER rovers)*
- ***Rice/USES (GSFC)***: *algorithm used in USES chip, with the multispectral predictor option.*
- ***JPEG-LS***: *is most efficient for 2D and is applied to the spectral bands independently*

Other Methods:

- **Differential JPEG-LS**: JPEG-LS applied to the differences between the successive spectral bands
- **SLSQ and SLSQ-OPT**: two versions of Spectral-oriented Least Squares (SLSQ) [Rizzo et al., 2005]. Algorithms with complexity roughly similar to that of ours.
- **3-D CALIC**: a nontrivial extension of the basic (2-D) CALIC algorithm to multispectral imagery. More complex.
- **M-CALIC**: multiband CALIC, another extension of CALIC to multispectral imagery. More complex.
- **ASAP**: Adaptive Selection of Adaptive Predictors [Aiazzi et al., 2001]; more computationally intensive than any of the other compressors in the tables

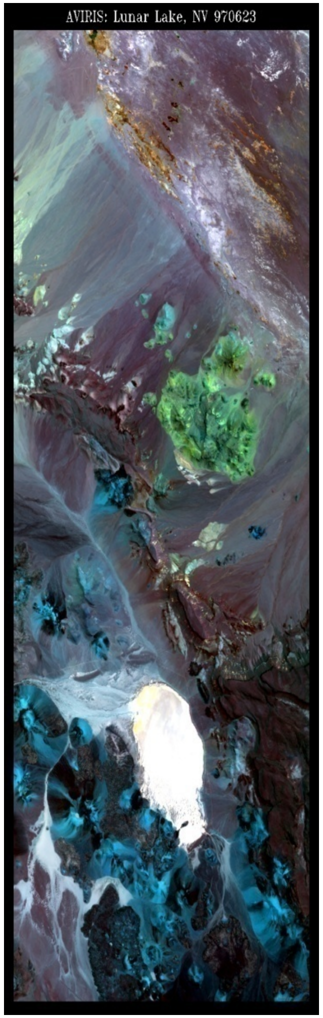
Comparison using Aviris Data Sets Test Bed

Moffett Field
(vegetation,
urban, water)

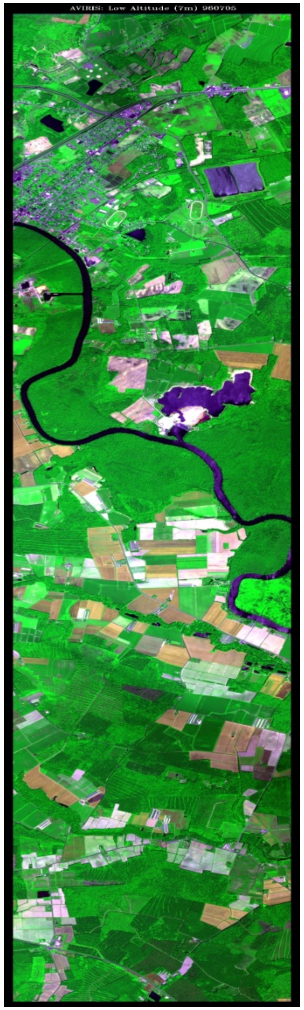


Cuprite
(geological
features)

Jasper Ridge
(vegetation)



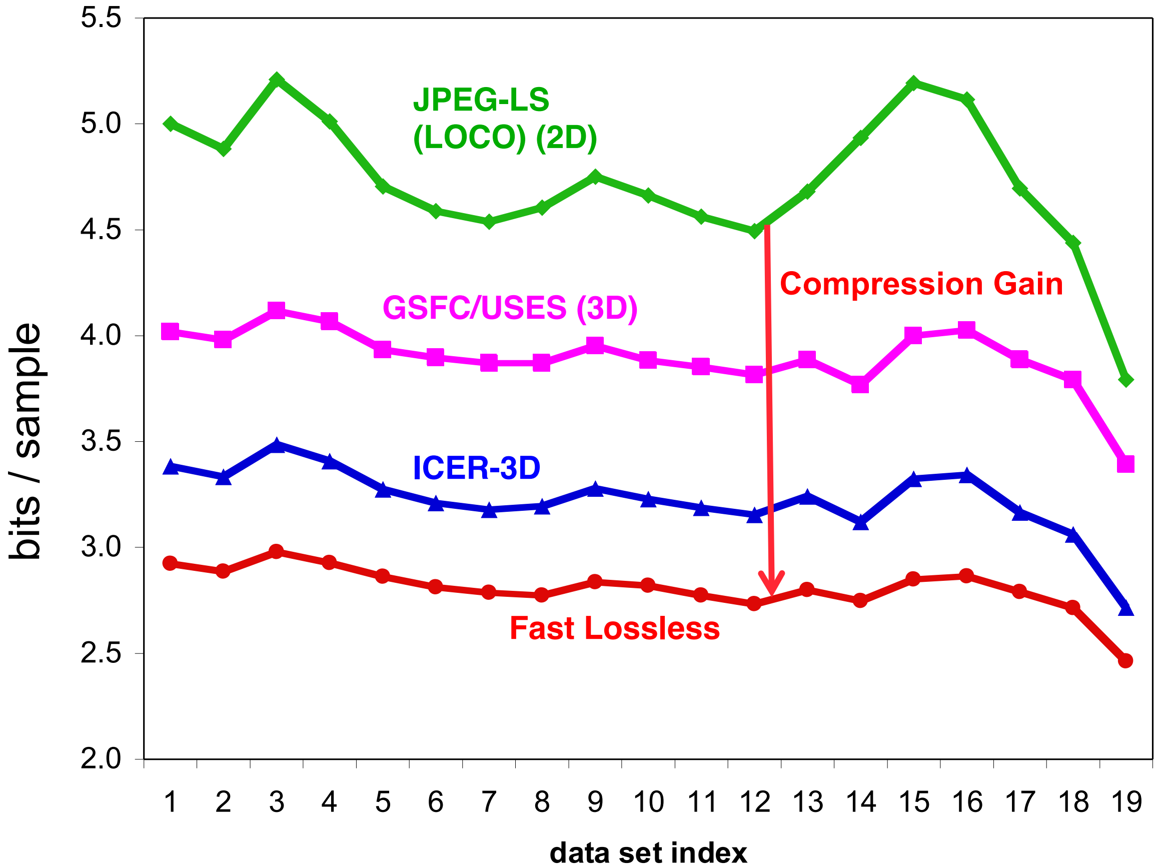
Lunar Lake
(calibration)



Low Altitude
(high spatial
resolution)

AVIRIS data sets represent different scenes

Comparison for raw AVIRIS Data



Compressor	rate (bits/ sample)
JPEG-LS (2D)	4.73
GSFC/USES Multispectral	3.89
ICER-3D	3.23
Fast Lossless	**2.81**

Compression performance averaged over 19 uncalibrated AVIRIS hyperspectral test data sets.

About 40% lower bit rate than state-of-the-art 2D approach (GSFC/USES).

Tests using 19 uncalibrated AVIRIS data sets:

- original sample size: 12 bits/sample
- data size: (614 × 512) pixels × 224 bands

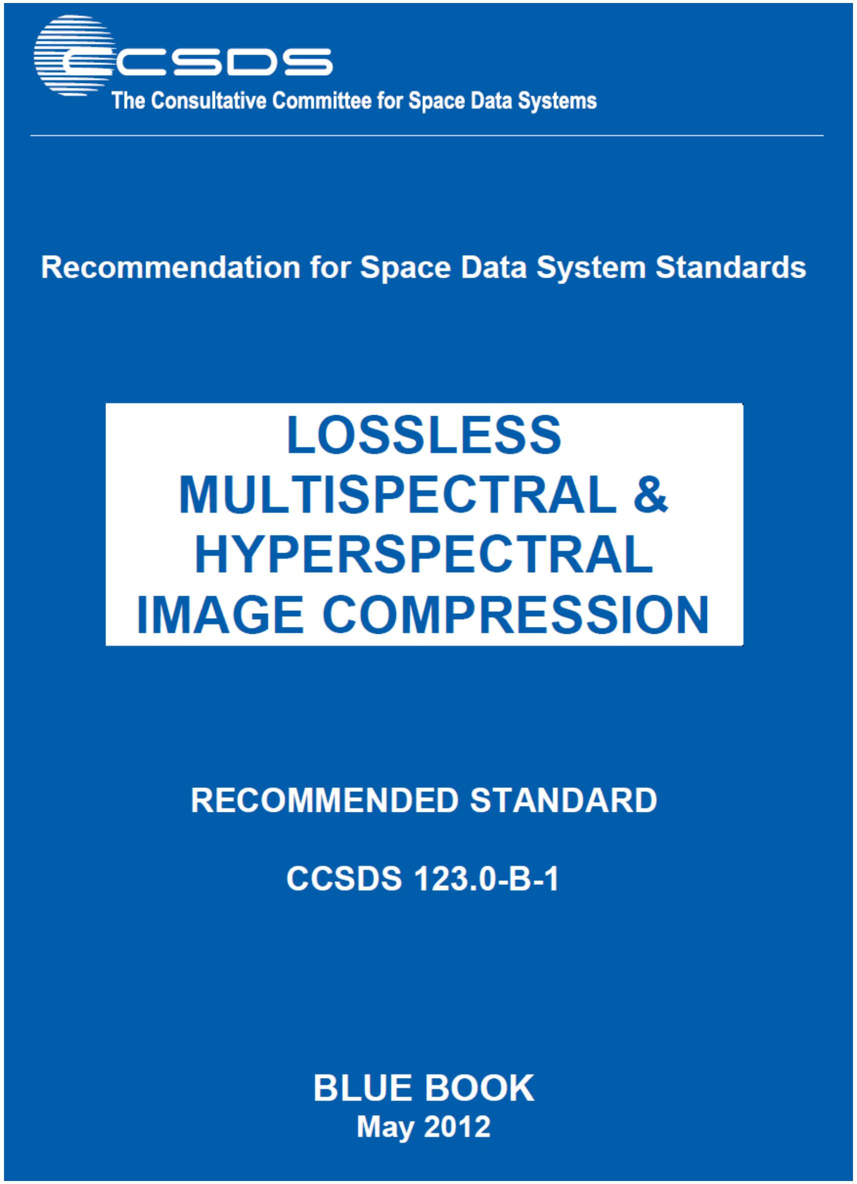
Methods:

JPEG-LS: is most efficient for 2D**; GSFC/USES** use chip; **ICER-3D** SOA (ICER-2D MER rovers)

Compression Algorithm Features

- **Performance:** outstanding compression effectiveness
- **Robust**; requires no training data or other specific information about the nature of the spectral bands for a fixed instrument dynamic range
- **Simple**: well-suited for implementation on FPGA hardware and easily parallelizable
- **Low computational complexity**. required operations per sample are:
  - 6 integer multiplications
  - 25 integer addition, subtraction, or bit shift operations
  - Golomb coding operations
- **Modest memory requirement**: enough to hold one spatial-spectral slice of the data (e.g., ≤650 Kbytes for AVIRISng data with 481 bands and 640 samples/line)
- **Instrument**: well-suited to push broom instruments

JPL Lossless Data Compression is a CCSDS Standard


CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS MULTISPECTRAL & HYPERSPECTRAL IMAGE COMPRESSION

RECOMMENDED STANDARD

CCSDS 123.0-B-1

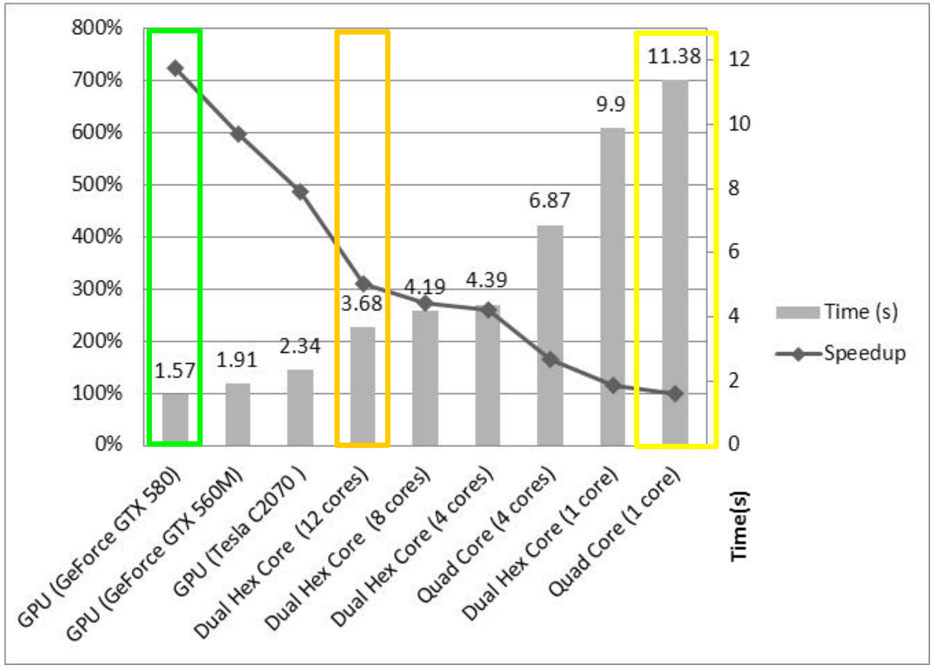
BLUE BOOK
May 2012

The Consultative Committee for Space Data Systems (CCSDS) Multispectral & Hyperspectral Data Compression working group has adopted the FL compressor as international standard CCSDS-123.0-B-1

FL verification software has demonstrated outstanding performance on all of the myriad airborne and spaceborne imagers represented in the CCSDS test data set:

- Hyperspectral imagers:
  AVIRIS, Hyperion, SFSI, CASI, M3, CRISM
- Ultraspectral sounders:
  AIRS, IASI
- Multispectral imagers:
  MODIS, MSG, PLEIADES, VEGETATION, SPOT5

High Speed FL Implementations: CPU/GPU



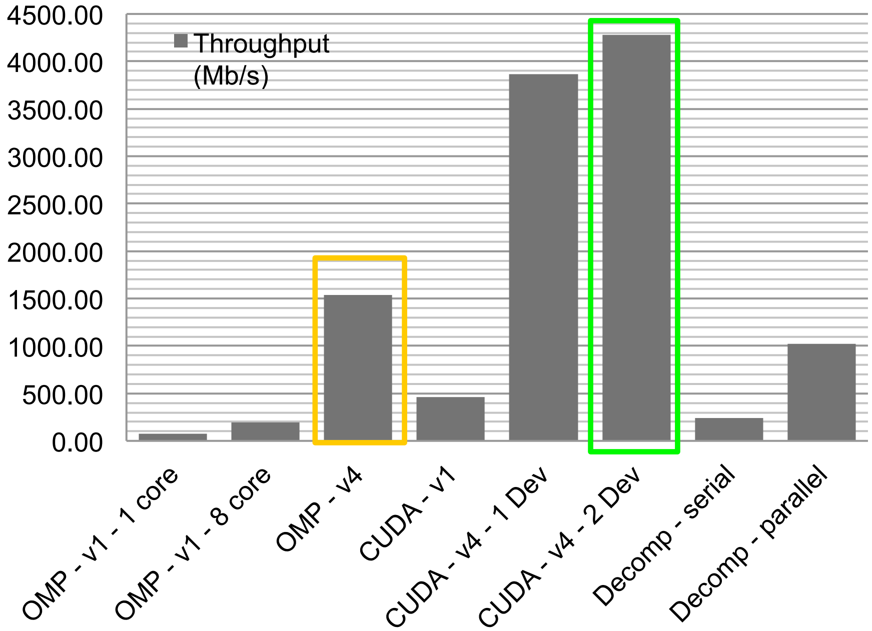
	Speedup	Time (s)	Speed (Mbit/s)	Speed (MSamp/s)
GPU GeForce GTX 580	725%	1.57	583.08	44.85
GPU GeForce GTX 560M	596%	1.91	479.29	36.87
GPU Tesla C2070	486%	2.34	391.21	30.09
Dual Hex Core (12 cores)	309%	3.68	248.76	19.14
Dual Hex Core (8 cores)	272%	4.19	218.48	16.81
Dual Hex Core (4 cores)	259%	4.39	208.53	16.04
Quad Core (4 cores)	196%	6.87	133.25	10.25
Dual Hex Core (1 core)	115%	9.9	92.47	7.11
Quad Core (1 core)	100%	11.38	80.44	6.19

Data Rate:
AVRISng (481*640 pixels per frames @100 frames/sec): 500Mbit/s
Future (481*1600 pixels per frames @100 frames/sec): 1300 Mbit/s
FPGA FL: 640 Mbit/s

- FL is well-suited for high-speed parallel implementations:
  - **GPU: 7× speed-up** – A GPU hardware implementation targeting the current state-of-the-art GPUs from NVIDIA®: mobile version GTX560M and desktop version GTX580
  - **OpenMP: 3× speed-up** – A 12-core implementation targeting the mobile Intel® quad-core i7™ processor and the desktop Intel® hexa-core Xeon™ processor
- Example: uncalibrated AVIRIS hyperspectral image (137MBytes)
  - Compression time: 11.38 sec on single-core CPU, 3.68 sec on 12-core CPU, and 1.57 sec on GPU

High Speed FL Implementations: CPU/GPU

Version 2: Even faster with re-designed data path



Version	Time (ms)	Throughput (Mb/s)	Throughput (MSamp/s)	Speedup vs. V1
OMP - v1 - 8 core	4488	194.53	14.96	1.00
OMP - v4 – 12 core	**569**	**1534.68**	**118.05**	**7.89**
CUDA - v1	1910	457.08	35.16	1.00
CUDA - v4 - 1 GPU	**226**	**3862.97**	**297.15**	**8.45**
CUDA - v4 - 2 GPU	**204**	**4279.56**	**329.20**	**9.36**
Decompress (serial)	3585	243.53	18.73	1.00
Decompress (parallel)	**857**	**1018.16**	**78.32**	**4.18**

Data Rate:
AVRISng (481*640 pixels per frames @100 frames/sec): 500Mbit/s
Future (481*1600 pixels per frames @100 frames/sec): 1300 Mbit/s
FPGA FL: 640 Mbit/s

- Redesigned data path implementation: Parallel computation across multiple 32 frames of the full image
- Total speed-up for Version 2
  - **GPU: 56× speed-up**– 137MB AVIRIS image compression time: 204 ms (vs. 11.38 sec)
  - **12-core CPU: 20× speed-up**– 137MB AVIRIS image compression time: 569 ms (vs. 11.38 sec)
- True real-time performance (2×-5× real-time target of 800Mb/s or 50MSamples/sec) BUT require 100 Watt

FL FPGA: **ARTEMIS & AVIRIS-NG**

FL FPGA Compression IPs for whiskbroom and pushbroom imagers

- **Xilinx Virtex-4 Lab Demonstration for ARTEMIS**
  - Implemented on Xilinx Virtex4 ML401 prototype board.
  - 17 MB image data (32 frames) uploaded serially to 256 DDR SDRAM prior to compression

- **Xilinx Virtex-5 Real-Time Airborne Onboard Compression**
  - Implemented pushbroom compressor on COTS Virtex 5 (equivalent to V5 Rad-hard device). Compresses one sample every clock cycle, a speed of 40 MSample/sec
  - Implementation tested in National Instruments PXI environment which includes a PXIe-7962R board with Xilinx Virtex-5 SX50T and two 256MBytes DRAMs. The system is connected to the airborne AVIRIS-NG HSI instrument and provides real-time onboard compression

ML401 Board

NI PXIe-7962R

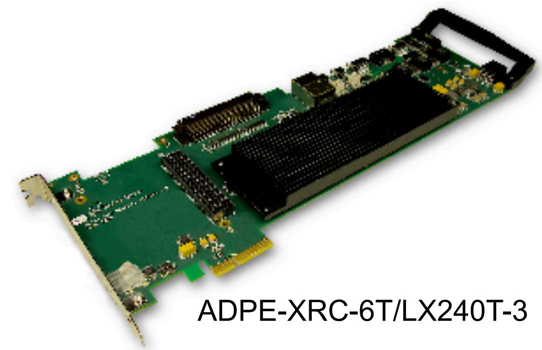
Twin Otter hosting AVIRIS-NG

14

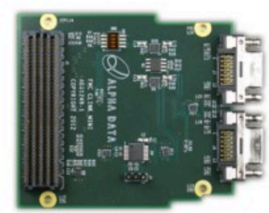
FL FPGA: PRISM & AVIRISng

Real-time aircraft onboard compression

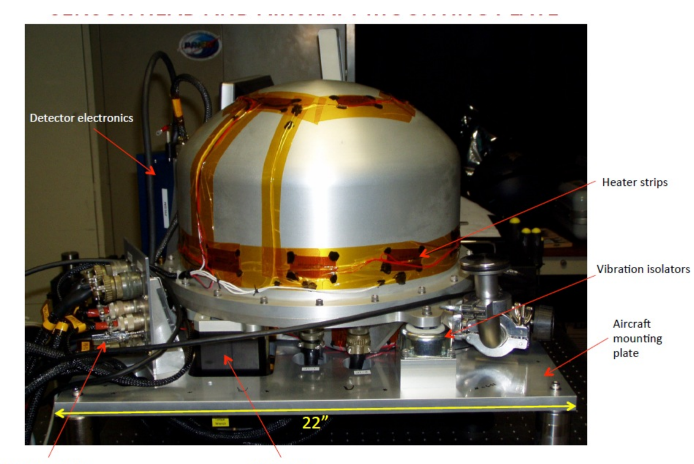
- Implemented pushbroom FL compressor on a COTS Virtex 6. Compresses one sample every clock cycle, a speed of 40 MSample/sec.
- Implementation tested via Alpha-Data ADPE-XRC-6T which includes
  - Xilinx Virtex-6 LX240T
  - two 256MBytes DRAMs (32bits data word, 3.2GBytes/sec per bank)
  - PCIe x4 Gen2 (500MBytes/sec per lane).
- PRISM and AVIRISng HSI image data transferred in real-time (60MBytes/sec) to the Virtex-6 via Alpha-Data FMC-CLINK-MINI camera link board, compressed on the Virtex-6 and transferred through PCIe to a 1GBytes SSD drive configured as RAID0 (500MBytes/sec)

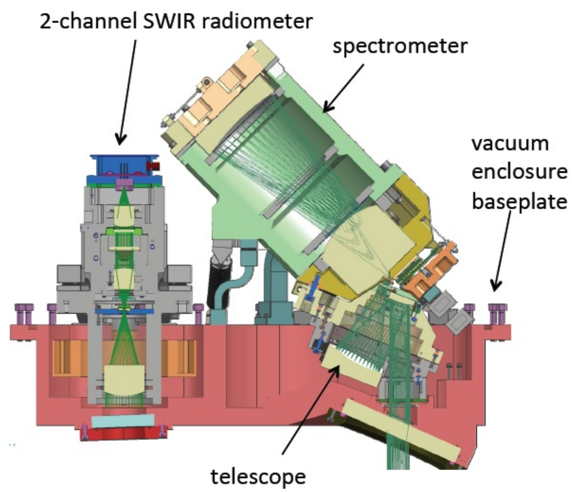


ADPE-XRC-6T/LX240T-3



FMC-CLINK-MINI

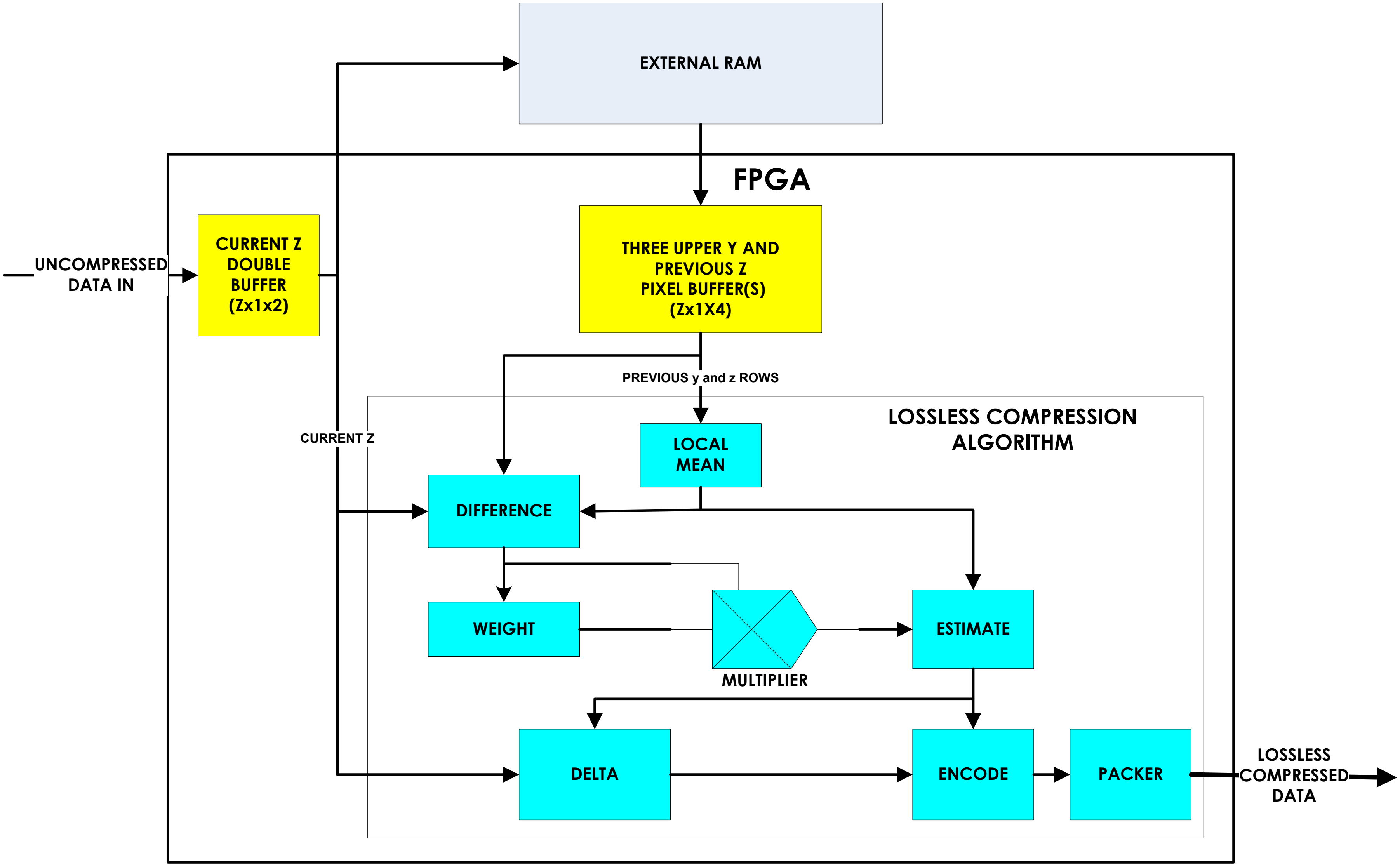




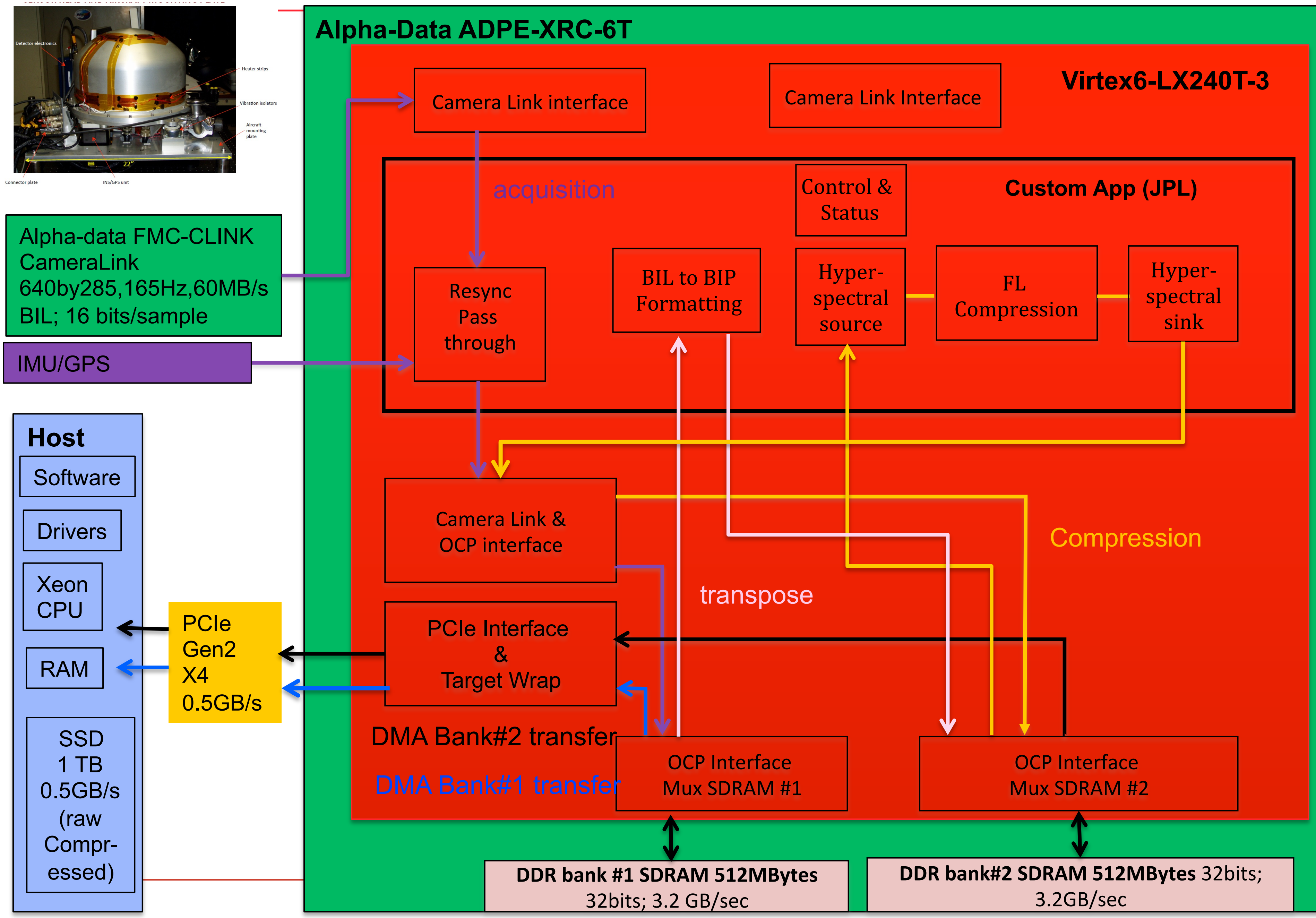
PRISM HSI

PRISM HSI Support Equipment

FL FPGA IP Main Block Diagram



FL FPGA Architecture



FL FPGA Resource Utilization – Virtex6

Device Utilization Virtex6-LX240T-3 (Compressor and Interface)

	Available	Used	Utilization All	Utilization Compressor	Utilization Virtex5 Compressor (estimate)
Slice Register (Flip-Flop)	301,440	37,284	12%	4%	8%
Slice Look-up-table (LUTs)	150,720	37,374	24%	8%	8%
Fully used LUT-Flip Flop pairs	50,693	19,105	38%	13%	26%
Block RAM/FIFO	416	108	25%	12%	12%
DSP 48eS	768	6	1%	1%	1%

Device Utilization SDRAM (AVIRISng)

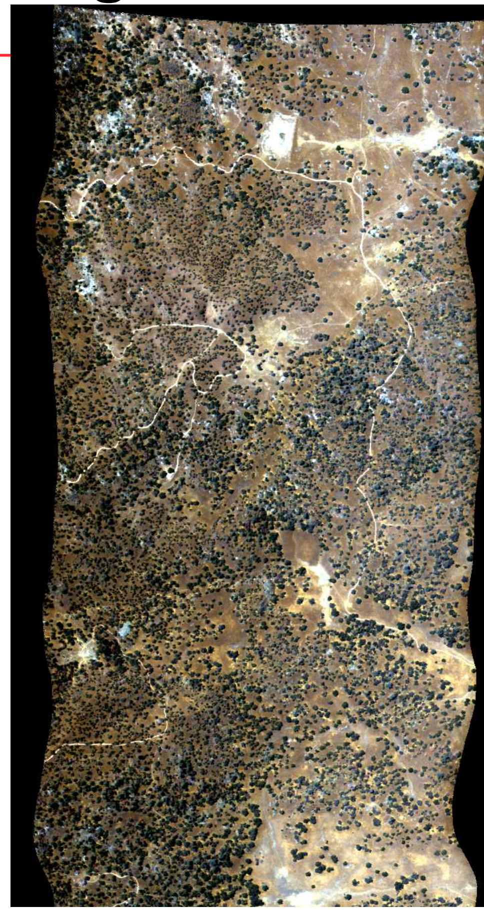
	Available	Used	Utilization
SDRAM Bank#1 (2 segments)	256 MBytes	40 MBytes	20 %
SDRAM Bank#2 (3 segments)	256 MBytes	60 MBytes	24 %

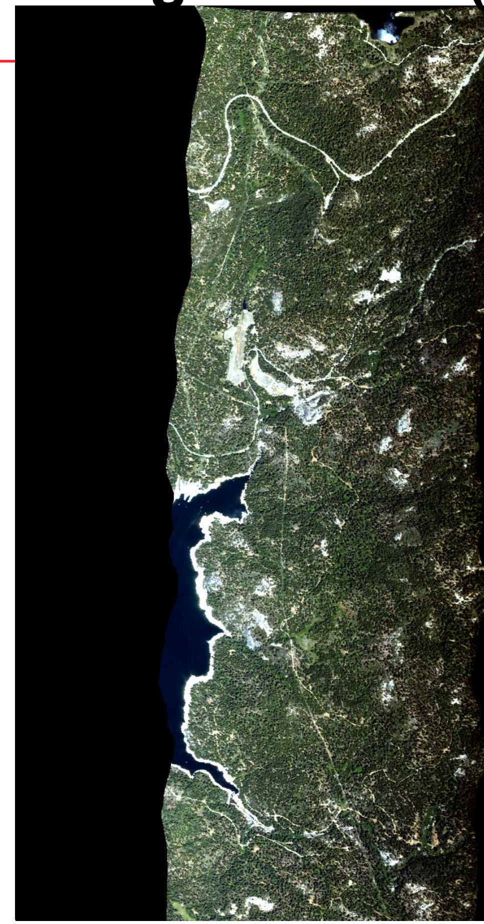
Timing: Critical Path

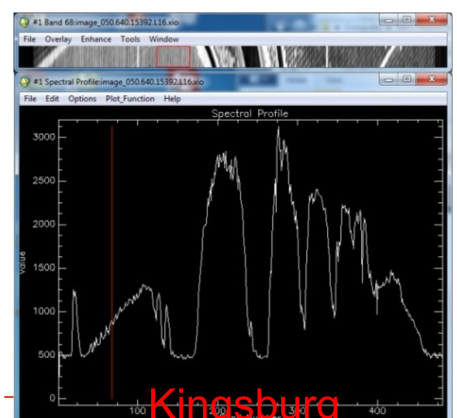
Block	Critical Path Timing
Synchronization frames with IMU/GPS	<25ns
Transpose BIP to BIL	<10ns
Predictor	12.070 ns
Entropy Encoder	10.029 ns
Packer	7.377 ns

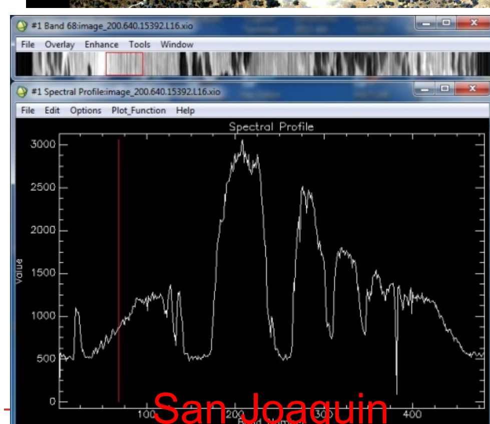
The implementation compresses one sample every clock cycle, which results in a speed of 40 MSample/sec

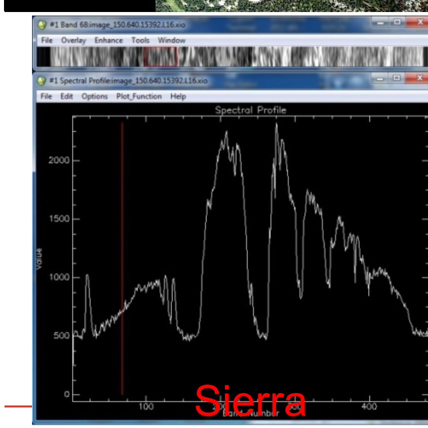
Comparison during airbone AVIRISng mission (June 2014)

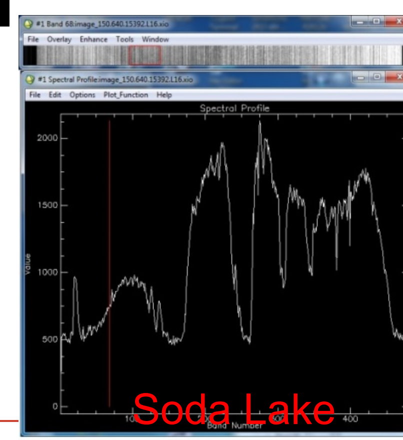




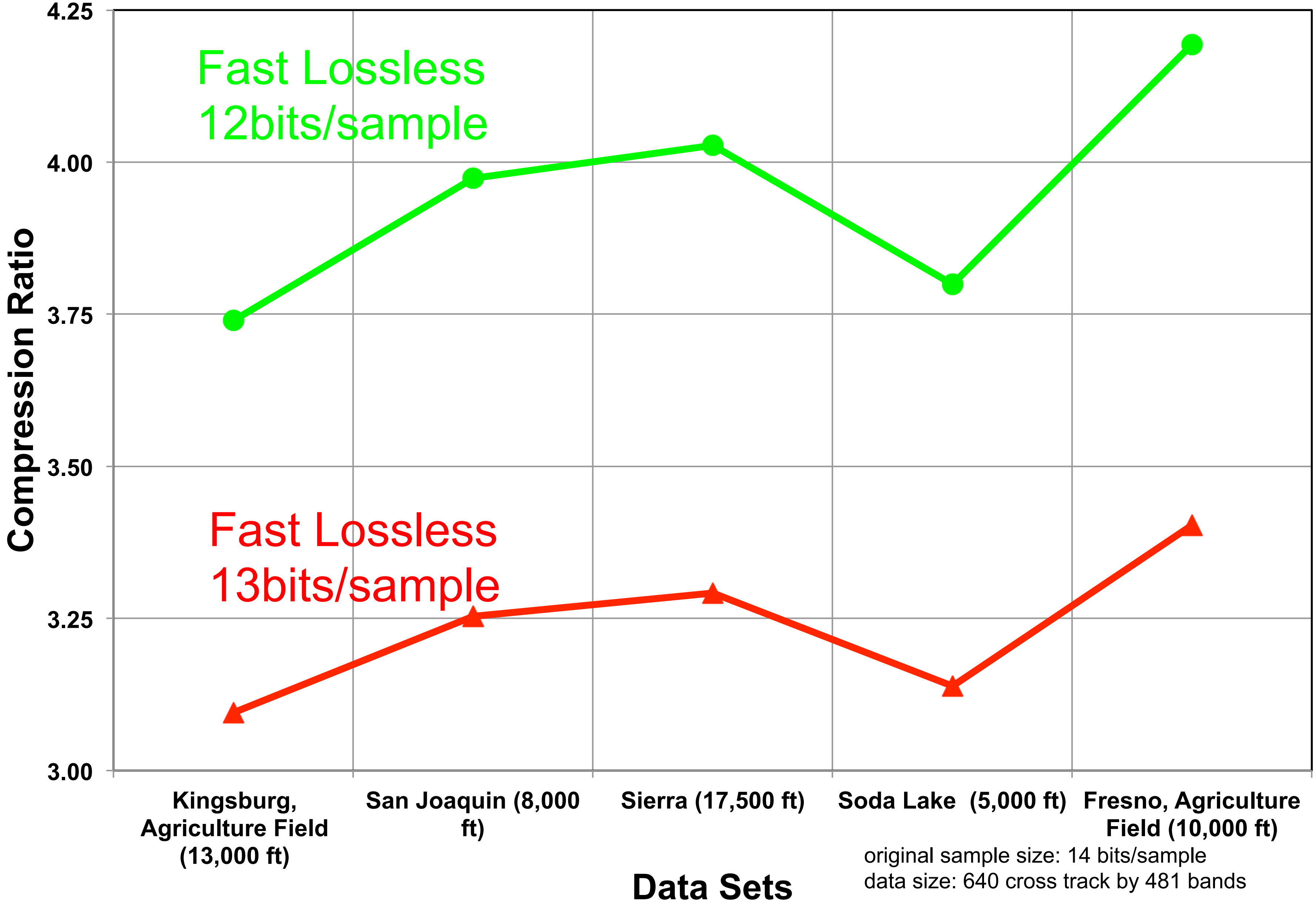








Comparison during airborne AVIRISng mission (June 2014)



original sample size: 14 bits/sample
data size: 640 cross track by 481 bands

Summary

We presented an FPGA implementation of a novel hyperspectral data compression algorithm and its flight demonstation: JPL adaptive Fast Lossless compressor.

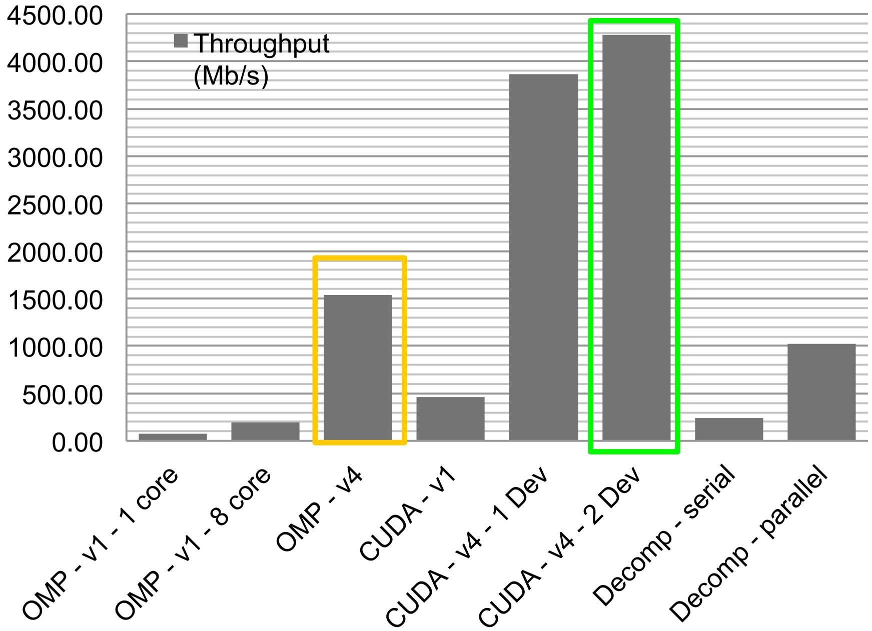
The implementation targets the Xilinx Virtex FPGAs and provides an acceleration of at least 7 times the software implementation on a single core of the Intel® Hex Core™ i7, making the use of this compressor practical for satellites and planet orbiting missions with hyperspectral instruments.

Future development will provide multiple implementations and near lossless data compression for accommodating large Focal Plane Array (FPA). We will also develop options to deploy various versions of the algorithm to accommodate data from different instrument types as well as radiance and reflectance data. And finally explore new hardware technologies such as System-on-the-Chip (SoC) to embed the compression next to the FPA ROI and fast I/O interface to the instrument (e.g. optical).

Back-up

High Speed FL Implementations: CPU/GPU

Version 2: Even faster with re-designed data path

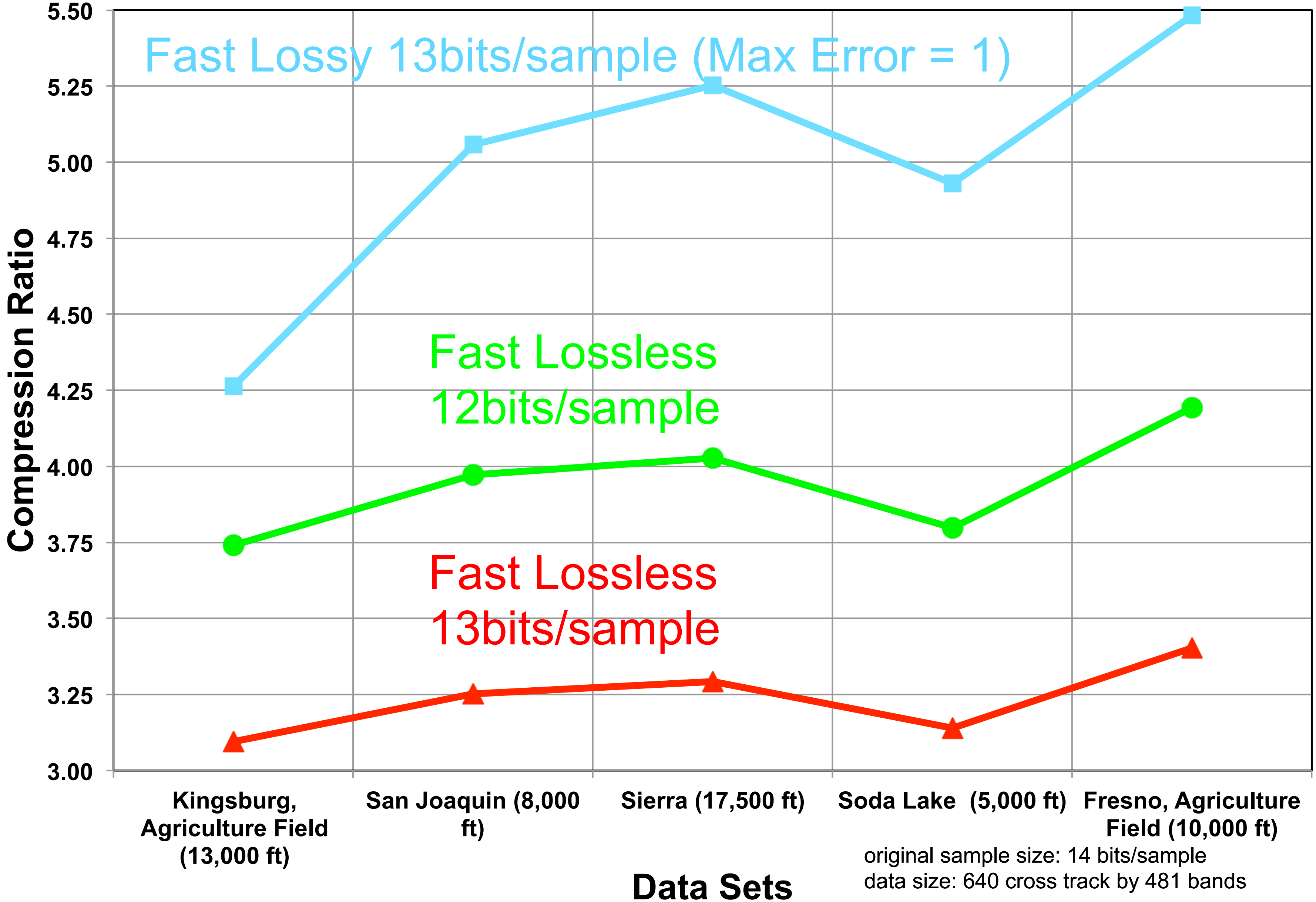


Version	Time (ms)	Throughput (Mb/s)	Throughput (MSamp/s)	Speedup vs. V1
OMP - v1 - 8 core	4488	194.53	14.96	1.00
OMP - v4 – 8 core	**569**	**1534.68**	**118.05**	**7.89**
CUDA - v1	1910	457.08	35.16	1.00
CUDA - v4 - 1 GPU	**226**	**3862.97**	**297.15**	**8.45**
CUDA - v4 - 2 GPU	**204**	**4279.56**	**329.20**	**9.36**
Decompress (serial)	3585	243.53	18.73	1.00
Decompress (parallel)	**857**	**1018.16**	**78.32**	**4.18**

Data Rate:
AVRISng (481*640 pixels per frames @100 frames/sec): 500Mbit/s
Future (481*1600 pixels per frames @100 frames/sec): 1300 Mbit/s
FPGA FL: 640 Mbit/s

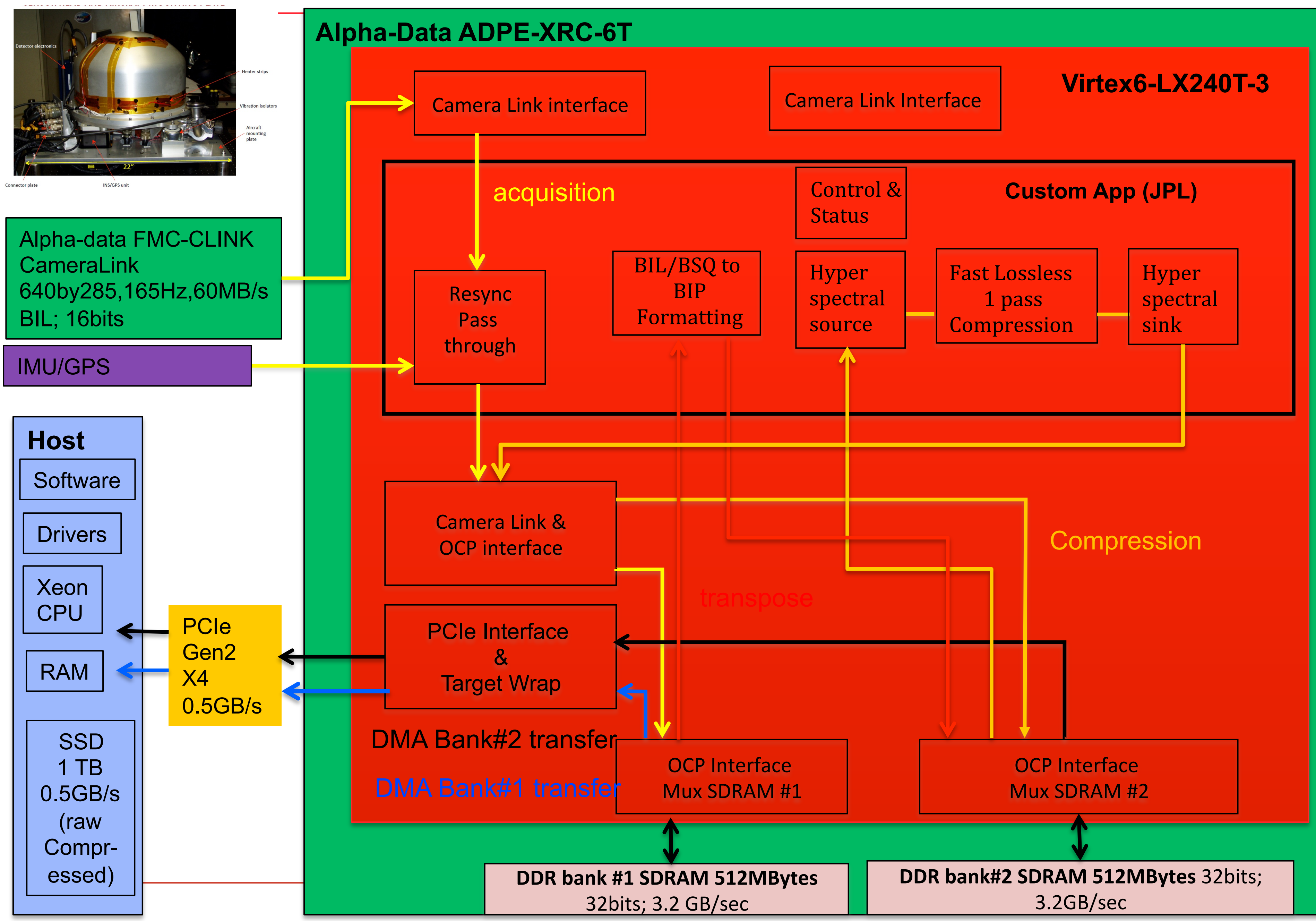
- Redesigned data path implementation:
  - Parallel computation across multiple 32 frames of the full image
  - Eliminated data writing to GPU main memory between algorithm stages
- Achieves further **8×** speedup for CUDA + OpenMP Implementations compared to Version 1
- Total speed-up for Version 2
  - **GPU: 56×** – 137MB AVIRIS image compression time: 204 ms (vs. 10 sec)
  - **12-core CPU: 24×** – 137MB AVIRIS image compression time: 569 ms (vs. 10 sec)
- Parallel Decompressor is 4× faster than serial
- True real-time performance (2×-5× real-time target of 800Mb/s or 50MSamples/sec)
- Supports multiple GPU cards

Comparison during airborne AVIRISng mission (June 2014)

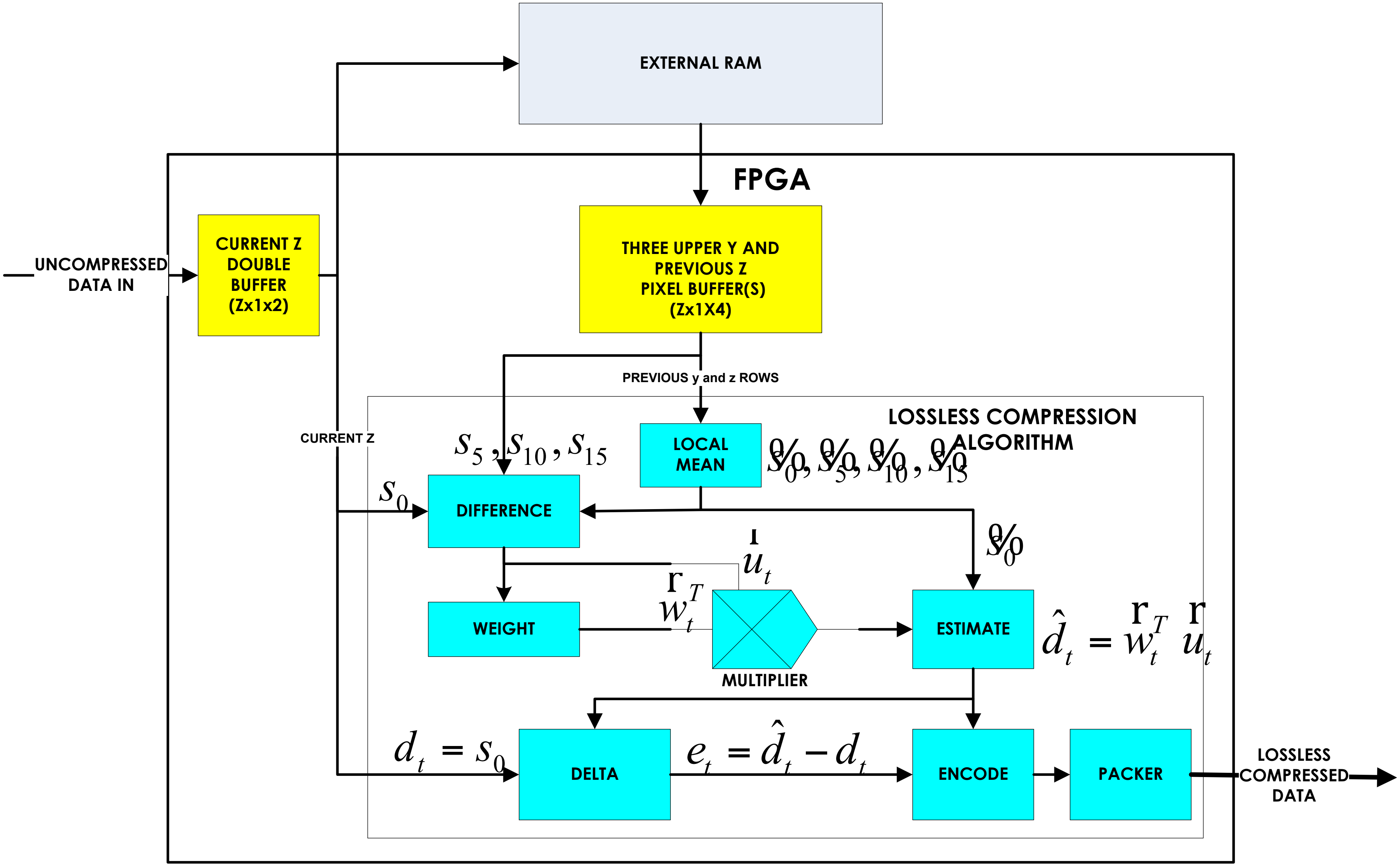


original sample size: 14 bits/sample
data size: 640 cross track by 481 bands

FL FPGA Architecture



FL FPGA IP Main Block Diagram

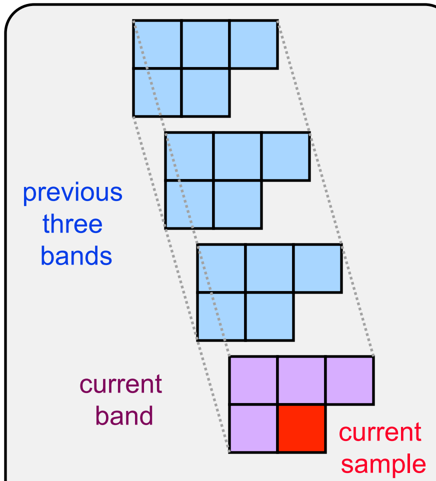


Fast Lossless Compression Algorithm

- **Objective:** State-of-the-art lossless compression, with low complexity (i.e., fast)
- **Approach**: *Predictive compression* that adapts to the data via the sign algorithm (a variation of the *least mean square (LMS) algorithm*) (see boxes below)
- **Compared** to *Transformed-based compression techniques* (such as DCT, Wavelet transform), this approach:
  - requires fewer arithmetic operations and less memory, simplifies data handling, and is more straightforward to implement (in software, DSP, or hardware)
  - yields significantly faster lossless compression
  - But provides only lossless (and potentially near-lossless) compression

Predictive Compression

- Encodes samples one-at-a-time, typically in raster scan order
- Estimates sample value probability distribution from previously encoded samples. These estimates are used to efficiently encode the sample value.
- The difference between an estimated sample value in the actual sample value is encoded in the compressed bitstream.



3D neighborhood used for prediction.

The *sign algorithm* and the *LMS algorithm* are members of a family of low complexity adaptive linear filtering techniques.

- Used extensively in signal processing applications
- Used for compression of audio data
- Not previously well studied for image or hyperspectral data compression

FL MSI/HSI Compressor

State of Development

- **Algorithm**
  - Described in published technical papers [1,2,3]
  - International standard for spacecraft onboard compression (next slide)
- **Software**
  - High speed parallel CPU multicore and GPU implementations [4]
- **Hardware**
  - FPGA lab hardware demonstration @ 33 MSamples/sec [5,6]
  - FPGA airborne demonstration @40 MSamples/sec with PRISM AVIRIS-NG

References:

[1] M. Klimesh, "Low-Complexity Lossless Compression of Hyperspectral Imagery Via Adaptive Filtering," *IPN Progress Report*, vol. 42-163, pp. 1–10, November 15, 2005.

[2] M. Klimesh, "Low-Complexity Adaptive Lossless Compression of Hyperspectral Imagery," (Invited paper), *SPIE 2006 Optics & Photonics Conference*, August 13-17, 2006, San Diego, CA; *Proc. SPIE*, vol. 6300, 9 pages, September 1, 2006.

[3] M. Klimesh, A. Kiely, P. Yeh, "Fast Lossless Compression of Multispectral and Hyperspectral Imagery," *Proc. 2nd Int'l Workshop on Onboard Payload Data Compression*, Toulouse, France, pp. 1–8, Oct. 28–29, 2010.

[4] D. Keymeulen, N. Aranki, B. Hopson, A. Kiely, M. Klimesh, K. Benkrid, "GPU Lossless Hyperspectral Data Compression System for Space Applications," *IEEE Aerospace Conference*, March 3-10, 2012, Big Sky, MT, USA

[5] N. Aranki, D. Keymeulen, M. Klimesh, A. Bakhshi, "Fast and Adaptive Lossless On-Board Hyperspectral Data Compression System for Space Applications," *2009 IEEE Aerospace Conference*, 8 pages, March 7-14, 2009, Big Sky, MT, USA.

[6] N. Aranki, D. Keymeulen, M. Klimesh, A. Bakhshi, "Hardware Implementation of Lossless Adaptive and Scalable Hyperspectral Data Compression for Space," *NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 315–322, July, 2009, San Francisco, CA, USA.

CCSDS Standardization of FL

The Consultative Committee for Space Data Systems (CCSDS) Multispectral & Hyperspectral Data Compression working group has adopted the FL compressor as international standard CCSDS-123.0-B-1 [7].

- FL verification software has demonstrated outstanding performance on all of the myriad airborne and spaceborne imagers represented in the CCSDS test data set:
  - Hyperspectral imagers:
    - AVIRIS, Hyperion, SFSI, CASI, M3, CRISM
  - Ultraspectral sounders:
    - AIRS, IASI
  - Multispectral imagers:
    - MODIS, MSG, PLEIADES, VEGETATION, SPOT5

Reference:

[7] *Lossless Multispectral & Hyperspectral Image Compression*. Recommendation for Space Data System Standards, CCSDS 123.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 2012.

JPL Lossless Data Compression is a CCSDS Standard

CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS MULTISPECTRAL & HYPERSPECTRAL IMAGE COMPRESSION

RECOMMENDED STANDARD

CCSDS 123.0-B-1

BLUE BOOK
May 2012

CCSDS RECOMMENDED STANDARD FOR LOSSLESS MULTISPECTRAL & HYPERSPECTRAL IMAGE COMPRESSION

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

– Agenzia Spaziale Italiana (ASI)/Italy.
– Canadian Space Agency (CSA)/Canada.
– Centre National d'Etudes Spatiales (CNES)/France.
– China National Space Administration (CNSA)/People's Republic of China.
– Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
– European Space Agency (ESA)/Europe.
– Federal Space Agency (FSA)/Russian Federation.
– Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
– Japan Aerospace Exploration Agency (JAXA)/Japan.
– National Aeronautics and Space Administration (NASA)/USA.
– UK Space Agency/United Kingdom.

Observer Agencies

– Austrian Space Agency (ASA)/Austria.
– Belgian Federal Science Policy Office (BFSPO)/Belgium.
– Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
– China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
– Chinese Academy of Sciences (CAS)/China.
– Chinese Academy of Space Technology (CAST)/China.
– Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
– CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
– Danish National Space Center (DNSC)/Denmark.
– Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
– European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
– European Telecommunications Satellite Organization (EUTELSAT)/Europe.
– Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
– Hellenic National Space Committee (HNSC)/Greece.
– Indian Space Research Organization (ISRO)/India.
– Institute of Space Research (IKI)/Russian Federation.
– KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
– Korea Aerospace Research Institute (KARI)/Korea.
– Ministry of Communications (MOC)/Israel.
– National Institute of Information and Communications Technology (NICT)/Japan.
– National Oceanic and Atmospheric Administration (NOAA)/USA.
– National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
– National Space Organization (NSPO)/Chinese Taipei.
– Naval Center for Space Technology (NCST)/USA.
– Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
– Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
– Swedish Space Corporation (SSC)/Sweden.
– United States Geological Survey (USGS)/USA.

FL Parameters & Options

FL Compression Parameters and Options:

- Prediction modes:
  - “regular”: for calibrated images and whisk-broom imagers
  - “pushbroom”: for raw images from pushbroom imagers (to handle detector artifacts)
- Number of previous spectral bands used for prediction, P
  - P=3 is typical. Increasing P leads to better but slower compression.
- Segment height (number of frames per segment)
  - Larger segments provide better compression because compressor has more time to adapt to image statistics.
  - Smaller segments provide better robustness to data loss and easier “random access” to portions of the data.
  - Because segments are compressed independently, this provides a simple method of exploiting parallelism
- Adaptation parameters
  - Prediction weight adaptation rate (determines how quickly prediction weights adapt to changing source statistics)
  - Entropy coding adaptation interval (determines how quickly entropy coder adapts to changing predictor accuracy)
- Segment initialization
  - Initial prediction weights can be tailored for a specific instrument
  - For raw images, detector offset array can be used to improve compression of initial line of each segment

- A good set of “default” parameter settings have been developed in the course of evaluation on the many different test images in the CCSDS test images

FL FPGA Resource Utilization – Virtex6

Device Utilization Virtex6-LX240T-3 (Compressor and Interface)

	Available	Used	Utilization
Slice Register	301,440	38,472	12%
Slice Look-up-table (LUTs)	150,720	48,115	31%
Fully used LUT-Flip Flop pairs	68,547	18,040	26%
Block RAM/FIFO	416	112	26%
DSP 48eS	768	9	1%

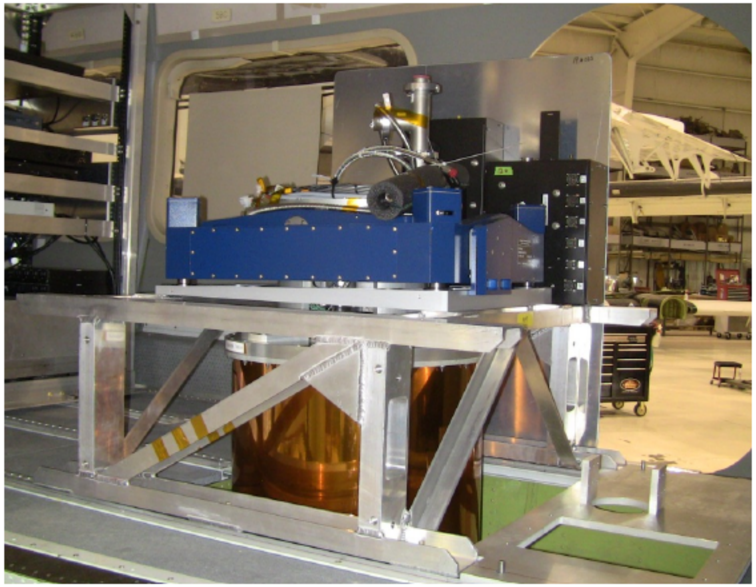
Device Utilization SDRAM

	Available	Used	Utilization
SDRAM Bank#1 (2 segments)	256 MBytes	24 MBytes	10 %
SDRAM Bank#2 (3 segments)	256 MBytes	36 MBytes	15 %

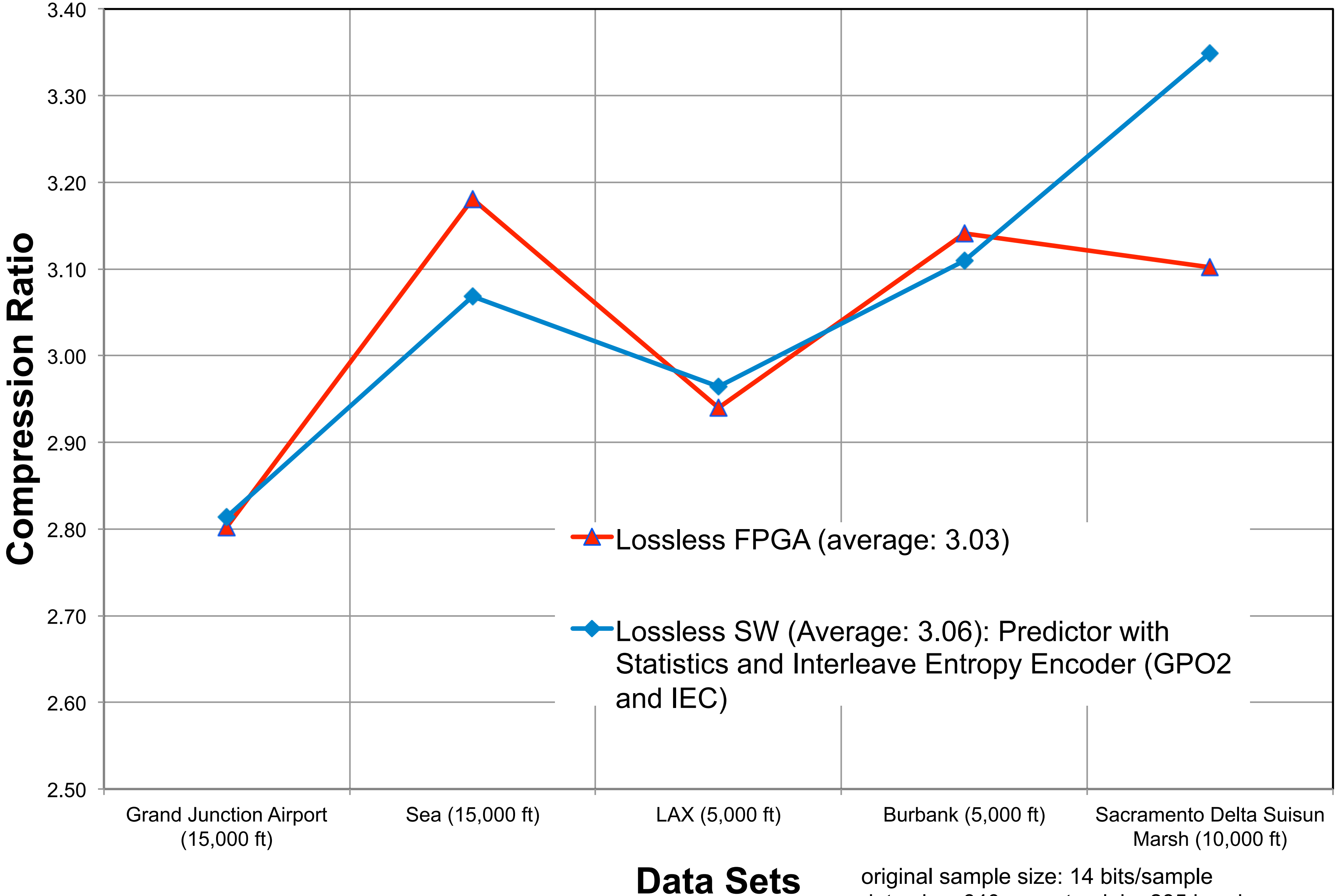
Timing: Critical Path

Block	Critical Path Timing
Synchronization with IMU/GPS	
Transpose BIP to BIL	
Predictor	12.033 ns
Entropy Encoder	10.029 ns
Packer	7.377 ns

FL Data Compression FPGA 2014 Flight Test of on PRISM and AVIRISng

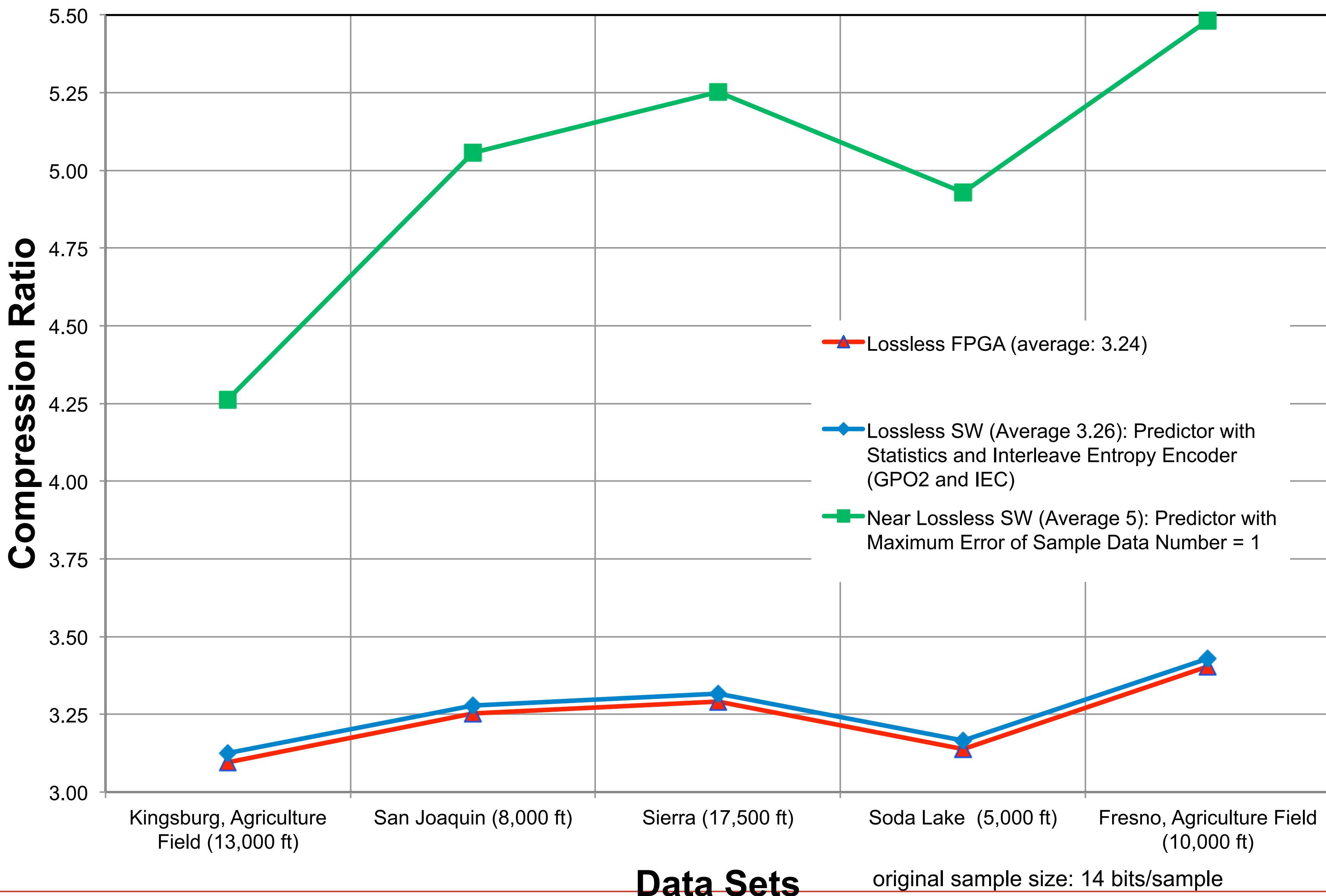


Comparison for raw PRISM Data (13 bits sample)



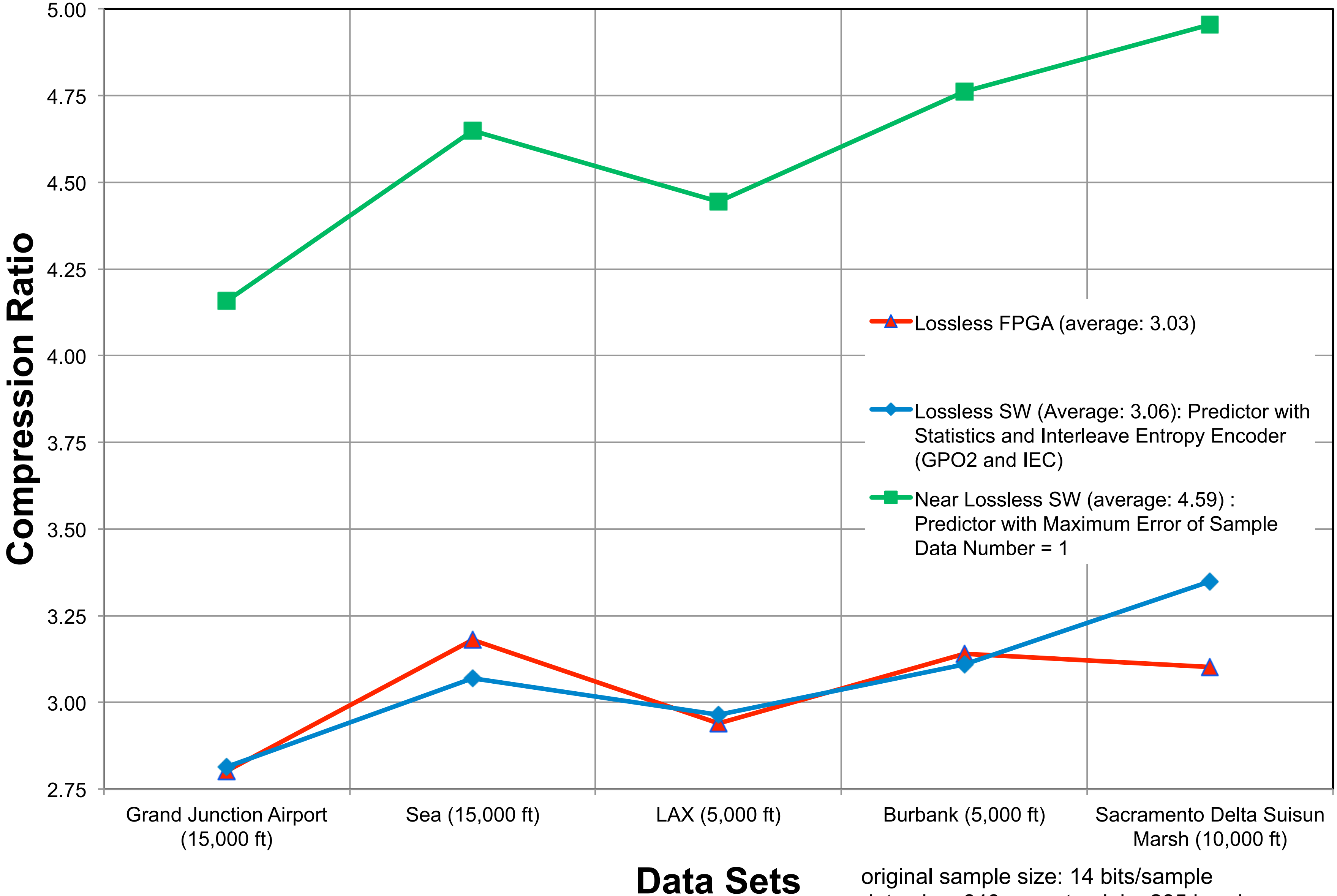
original sample size: 14 bits/sample
data size: 640 cross track by 285 bands

Comparison for raw AVIRIS Data (13 bits sample)



original sample size: 14 bits/sample
data size: 640 cross track by 481 bands

Comparison for raw PRISM Data



original sample size: 14 bits/sample
data size: 640 cross track by 285 bands

Hardware Performance Summary

VIRTEX-5 SX50T Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Slice Registers	1586	32640	4%
Slices LUTs	12697	32640	38%
Block RAM/FIFO	8	132	6%
DSP4BEs	3	288	1%

VIRTEX-5 SX50T Timing & Power

Critical Path Delay: 24.29ns (41Mhz)

Total power: 702.20 mW

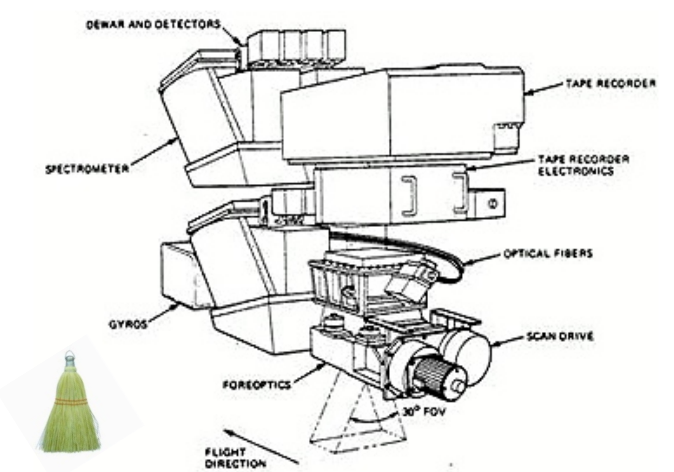
The implementation compresses one sample every clock cycle, which results in a speed of 40 MSample/sec

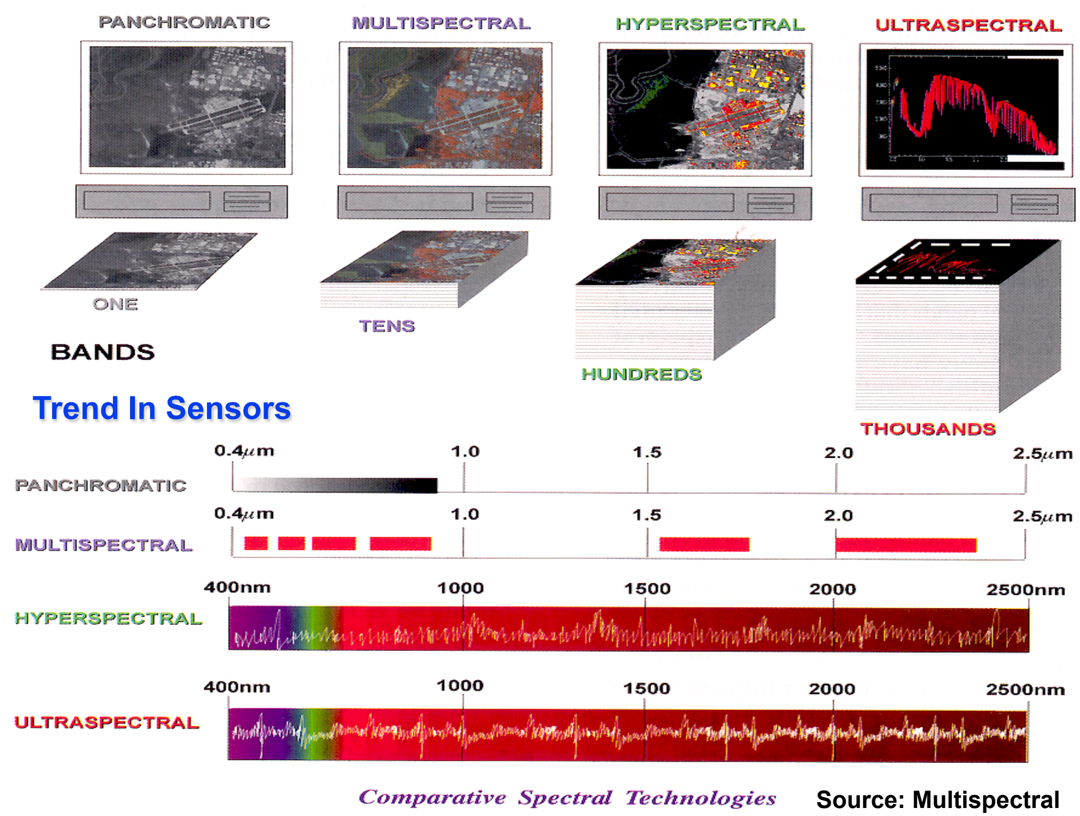
VIRTEX-5 SX50T and Space-qualified Rad Hard VIRTEX-5QVFX130

Resource Available	VIRTEX-5 SX50T	VIRTEX-5QV FX130
Slice Registers	32,640	122,880
Slice LUTs	32,640	122,880
Block RAM/FIFO	132	456
DSP48E	288	384

Hyperspectral Imager (AVIRIS)

- **Airborne Visible and Infrared Imaging Spectrometer (AVIRIS)** is a multispectral imagers with the same detector element for all samples in a given spectral band ("whisk broom"-type instrument).
  - **Spectral Resolution**: AVIRIS has 224 detectors (channels) in the spectral dimension, extending over a range of 380nm to 2500 nm.
  - **Spatial resolution**: A typical mission, mounting AVIRIS on a NASA aircraft (ER-2), produces a spatial resolution of about 20 meters, but can improve that to five meters by flying at lower altitudes





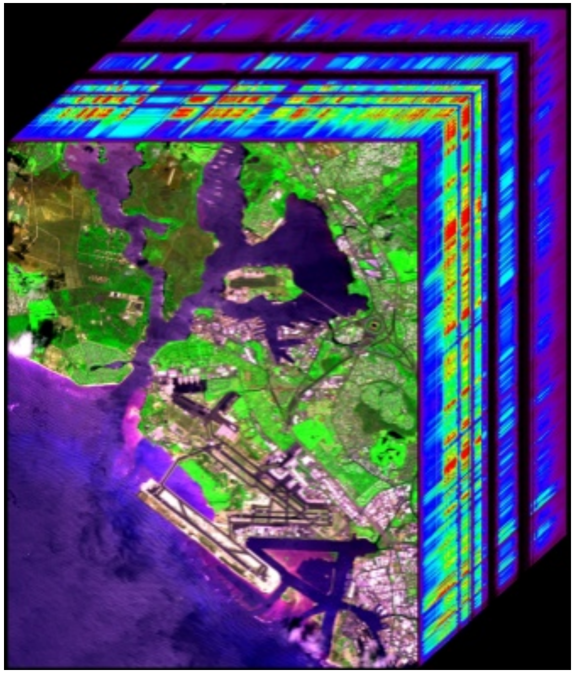
Source: Multispectral Imagery Guide, Logicon

Hyperspectral Images

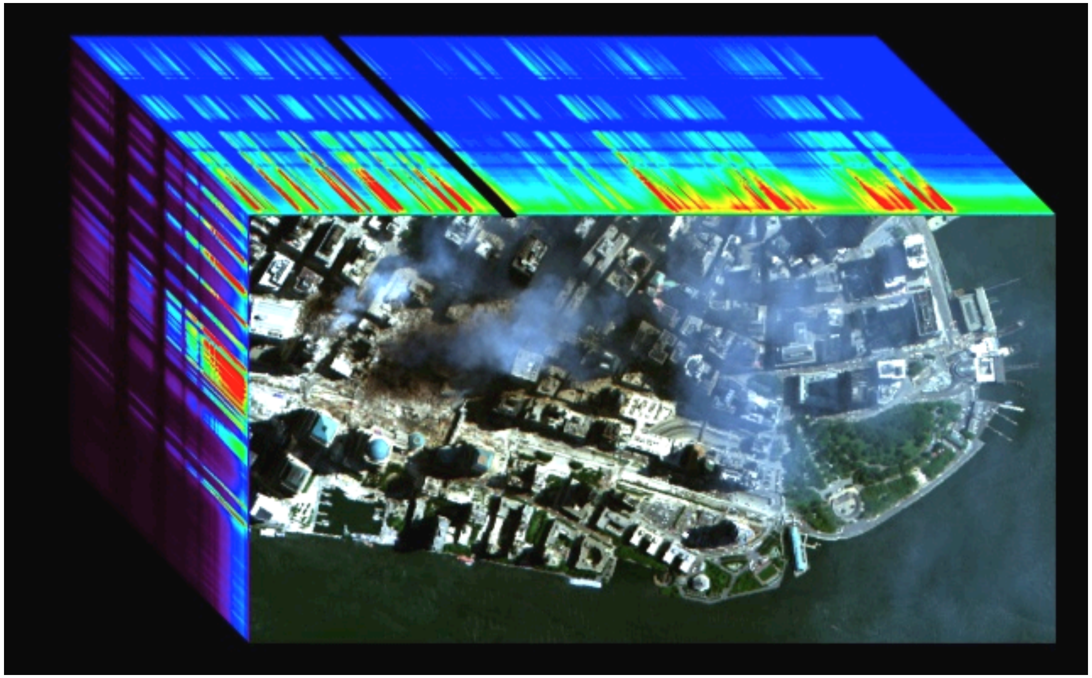
- **Hyperspectral images are three-dimensional data sets**, where two of the dimensions are spatial and the third is spectral.
  - A hyperspectral image can be regarded as a stack of individual images of the same spatial scene, with each such image representing the scene viewed in a narrow portion of the electromagnetic spectrum, referred to as spectral bands.
  - Hyperspectral images typically consist of hundreds of spectral bands;

  → The voluminous amount of data comprising hyperspectral images (up to 645GB) makes them appealing candidates for data compression.

AVIRIS hyperspectral data "cubes"



Pearl Harbor, Hawaii



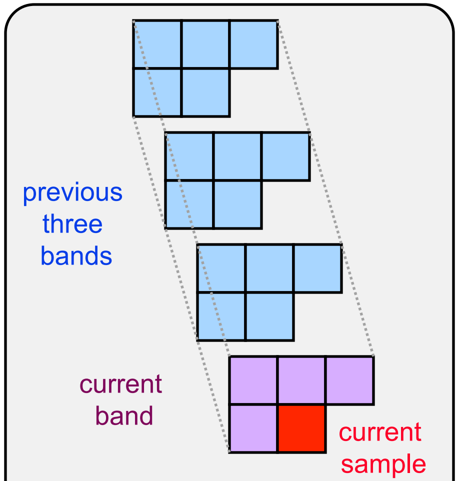
WTC Disaster Site

Fast Lossless Compression Algorithm

- **Objective:** State-of-the-art lossless compression, with low complexity (i.e., fast)
- **Approach**: *Predictive compression* that adapts to the data via the sign algorithm (a variation of the *least mean square (LMS) algorithm*) (see boxes below)
- **Compared** to *Transformed-based compression techniques* (such as DCT, Wavelet transform), this approach:
  - requires fewer arithmetic operations and less memory, simplifies data handling, and is more straightforward to implement (in software, DSP, or hardware)
  - yields significantly faster lossless compression
  - But provides only lossless (and potentially near-lossless) compression

Predictive Compression

- Encodes samples one-at-a-time, typically in raster scan order
- Estimates sample value probability distribution from previously encoded samples. These estimates are used to efficiently encode the sample value.
- The difference between an estimated sample value in the actual sample value is encoded in the compressed bitstream.



3D neighborhood used for prediction.

The *sign algorithm* and the *LMS algorithm* are members of a family of low complexity adaptive linear filtering techniques.

- Used extensively in signal processing applications
- Used for compression of audio data
- Not previously well studied for image or hyperspectral data compression

FLEX FPGA IPs Development

Capabilities:

- Consistency between FLEX software and FPGA compressors
- Handles BIP, BIL and BSQ input formats
- Floating point data handled by rounding to integers (Phase 1 FLEX) and/or new methods developed during Phase 2 effort (Algorithms Subtask 2.3)

IP Sub-Modules:

- Compute image statistics: new design
- Format BIL/BSQ to BIP: design to be modified from FL FPGA; native format for compression IP is BIP
- Predictor: design to be modified from FL FPGA
- Quantizer: new design
- Hybrid Entropy Coder: new design, will include GPO2 encoder developed for FL FPGA
- Packer with segment markers: design to be modified from FL FPGA
- Erasure Correction Encoder: new design

Software Driver for FPGA Implementation

Software Driver Tasks

- Interpret command-line parameters
- Acquire image parameters
  - E.g., in the case of an image saved in ENVI format, parse the ENVI header file to extract the image parameters
- Send compression parameters (including user-selected values and image parameters) to FPGA
- Generate text header for compressed file (e.g., file identifier text plus verbatim ENVI header) and send it to the FPGA board
  - FPGA needs this header because it is protected with parity words
  - Assumption: parity words must be generated on FPGA
- Read image file (from SSD) and send raw image data to FPGA board
- Receive compressed image data from FPGA board and write file to SSD

Notes

- Software driver memory requirement may be significantly smaller than image file size
  - Possible to read image data and send it to FPGA in chunks (large, but much smaller than whole image)
  - Similarly, compressed data can be received from FPGA and written to file in chunks
- Software drivers will use the alpha-data drivers which allow:
  - DMA from Host to FPGA board SDRAM (raw image)
  - DMA from FPGA board SDRAM to Host (compressed image)
  - Read/Write internal FPGA registers
  - Interrupt Handler to initiate DMA transfer independently of data processing on the FPGA board

FPGA Implementation Trade-Offs

- Large on-board SDRAM memory is required (min 2 Gbytes)
  - Needed to support 2-pass compression approach, in which the first pass computes statistics over the whole image.

  Alternative approach: Compress image in chunks, which will reduce SDRAM memory needed. This would lead to some variations in quality between chunks when compression is done to a compression ratio target.

- Serial implementation may not meet RDUCE latency objective
  - However, level of pipelining is still to be determined for the FPGA implementation components such as reading image file through DMA, computing statistics, data formatting, compression, and writing compressed data back through DMA.

  Alternative approach: Parallel implementation at segment level; this would require more resources and power.

- Handling BSQ data format may introduce a latency of approximately 2 seconds
  - Due to the nature of BSQ images, formatting to BIP may require reading full image into local board SDRAM prior to starting formatting operation.

  Alternative approach: Eliminate the capability to handle BSQ input

- Innovative algorithmic enhancements may have to be abandoned if they cannot be implemented in hardware on schedule

  Alternative approach: Delay HW delivery schedule by 6 months (to month 24 from start) to allow migration of such enhancements to HW.

Need few Slides with Hardware Implementation Performance

High Speed FL Implementations: FPGA

Hardware Performance Summary

Virtex-5 SX50T

Device Utilization

	Available	Used	Utilization
Slice Register	32,640	1,586	4%
Slice Look-up-table	32,640	12,697	38%
Fully used LUT-FF pairs	13,385	898	6%
Block RAM/FIFO	132	8	6%
BUFG/BUFGCTRLs	32	1	3%
DSP 48eS	288	3	1%

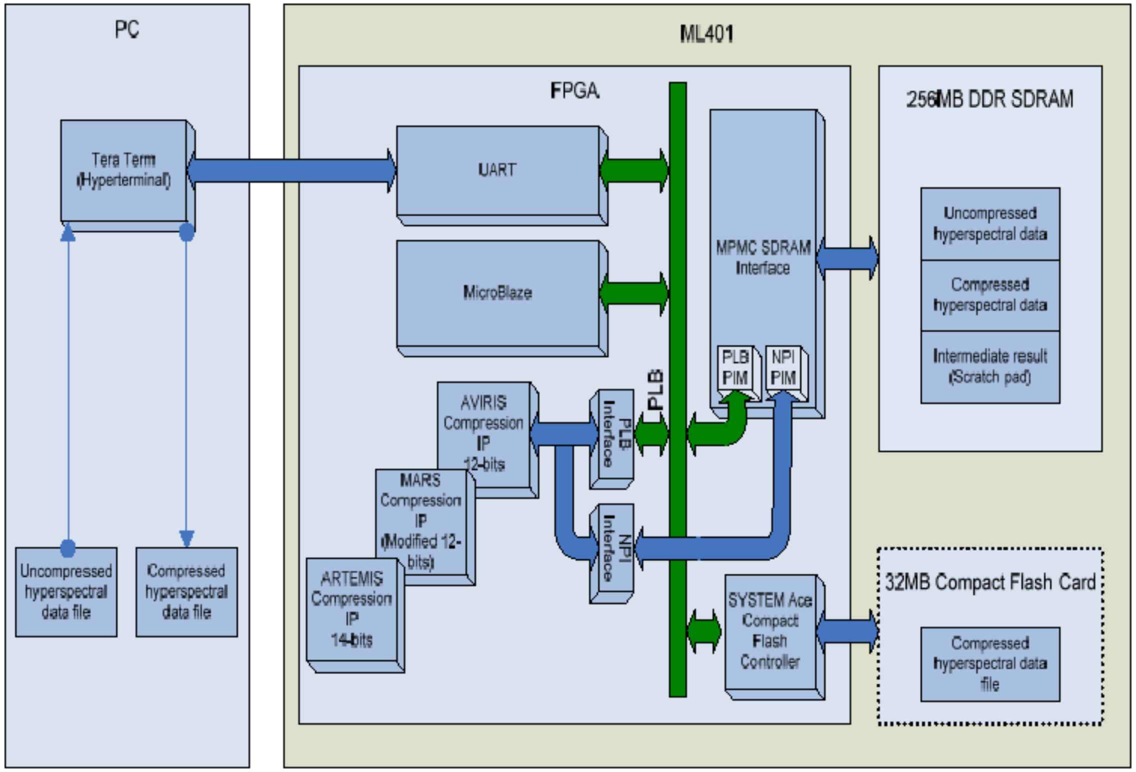
Timing & Power

- Delay (ns) (41Mhz) 24.29
- Total Power 700mW

The implementation compresses one sample every clock cycle, yielding a speed of 41 MSample/sec

JPL Compression IP integrated into ARTEMIS

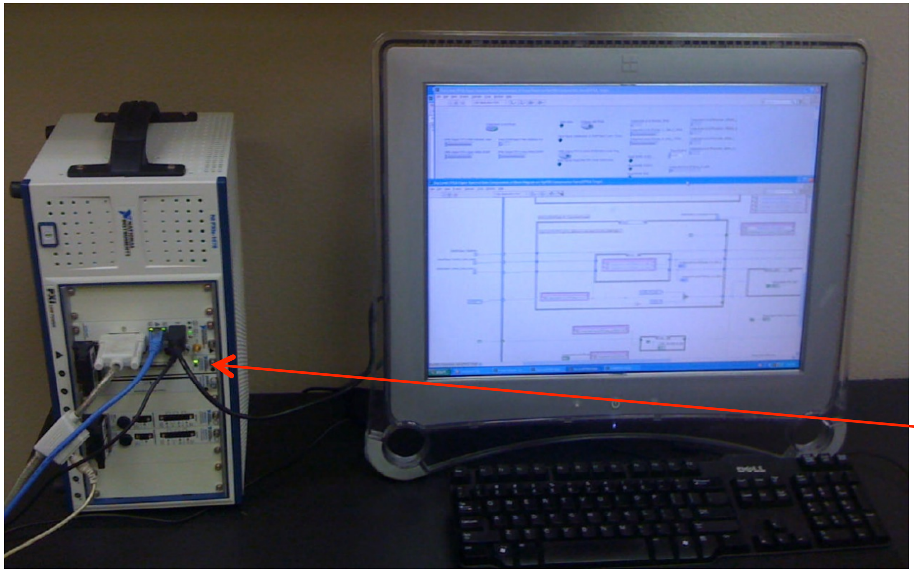
JPL Fast lossless Compression IP is being implemented on the ML401 Virtex4 Xilinx prototype boar



JPL Fast Lossless Compression IP will be integrated into Advanced Responsive Tactically-Effective Military Imaging Spectrometer (ARTEMIS) payload which consists of a telescope, a high resolution pushbroom imaging spectrometer, a high resolution imager and a real-time processor.

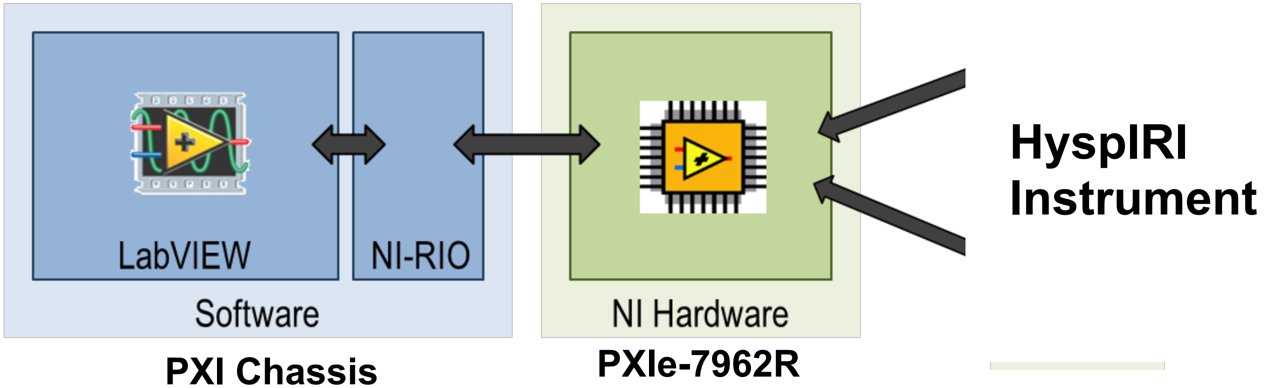
JPL Compression IP integrated into AVIRISng

JPL Fast lossless Compression IP is currently being implemented on the National instrument PXI environment which includes a PXI chassis and PXIe-7962R hardware with Xilinx Virtex-5 SX50T and two 256MBytes DRAMs



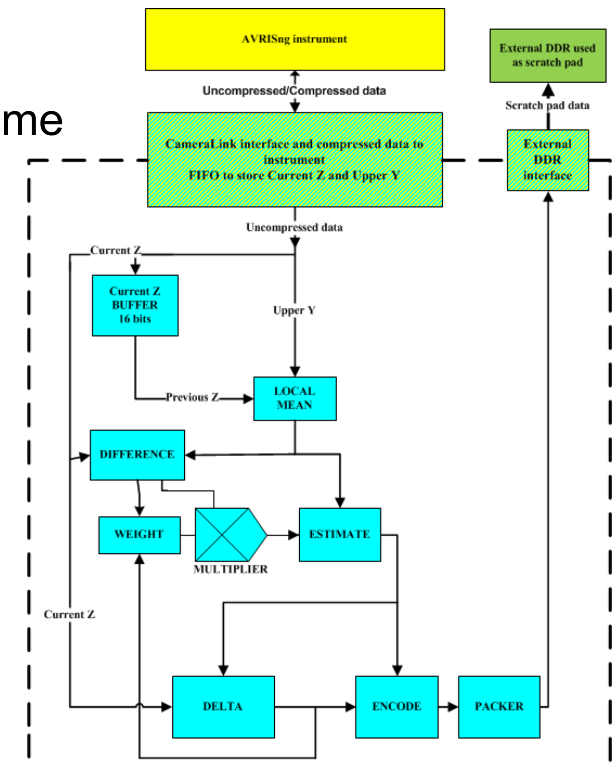
PXI National Instrument Testbed

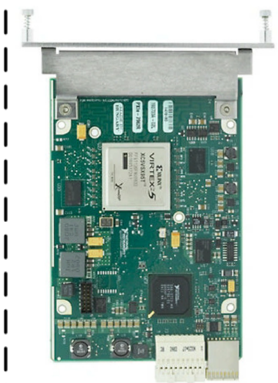
PXIe-7962R with Xilinx Virtex-5 SX50T



Real-time aircraft onboard FPGA compression

- Implemented on a commercial Virtex 5 (equivalent to V5 Rad-hard device). Compresses one sample every clock cycle, a speed of 40 MSample/sec with total power of 700 mW.
- FL compressor implementation tested in National Instruments PXI environment which includes a PXIe-7962R board with Xilinx Virtex-5 SX50T and two 256MBytes DRAMs. The system is connected to the airborne AVIRIS-NG HSI instrument and compresses HSI data in real-time on the plane.





48

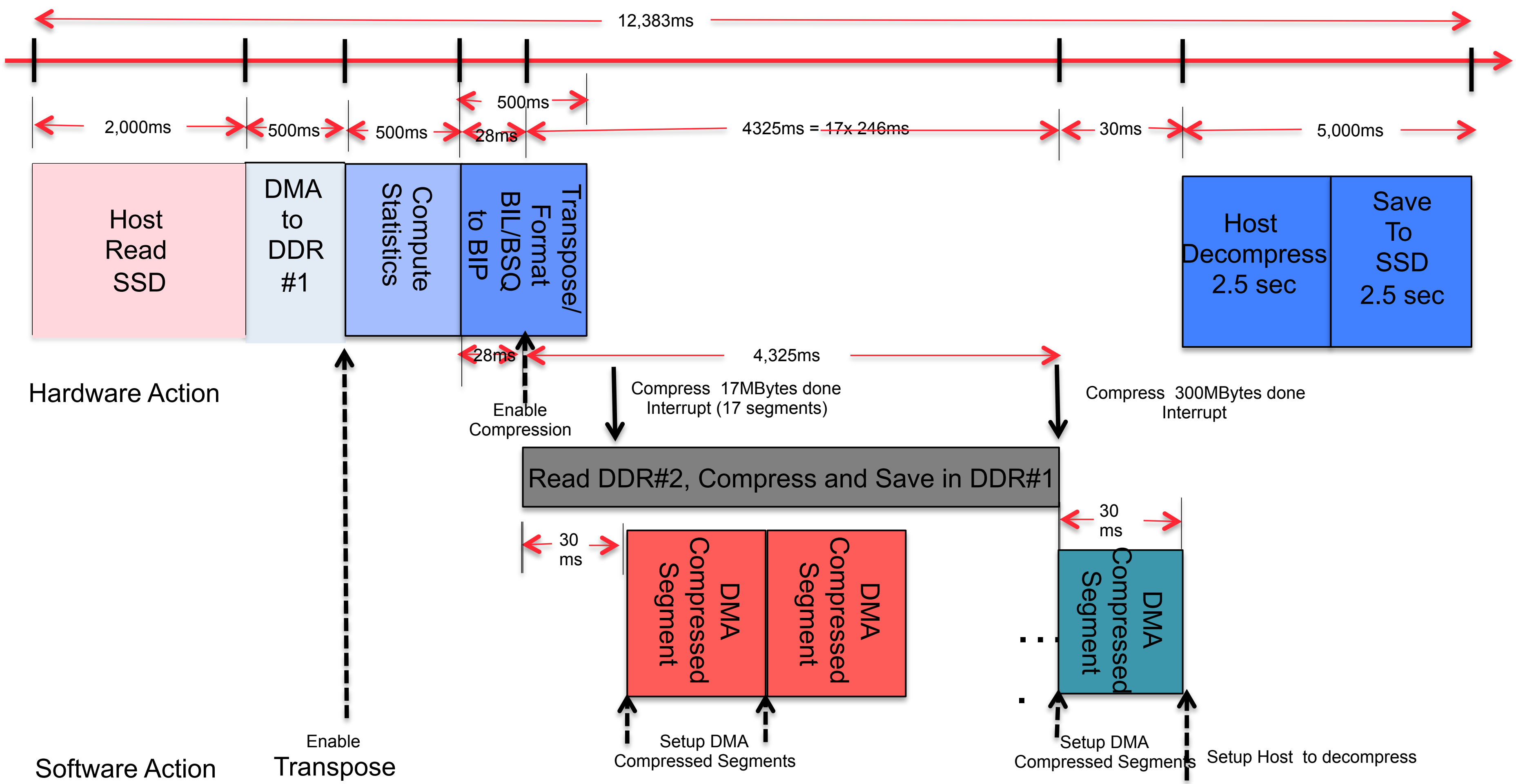
Need few Slides with Alpha Data Architecture

FLEX FPGA Timing (estimated)

Compress 300MB Image in < 15 sec

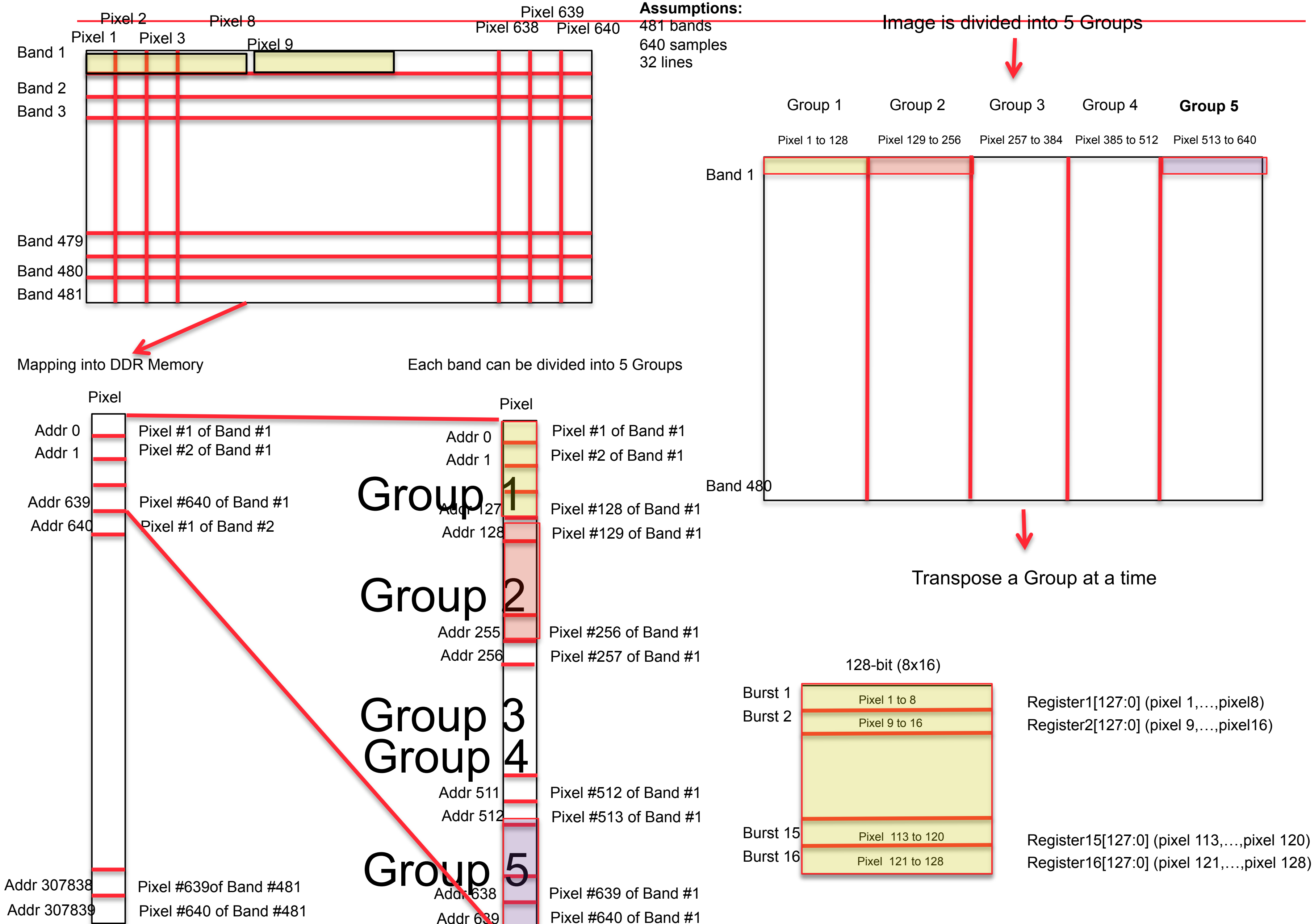
Assumptions:

- 32 frames/segment, 480×640 samples/frame, 16bits/sample
- Transpose/Format 1 segment in 30ms
- Compress 1 segment in 246ms (estimate based on FL)
- Does not include parity check: Block read of 1 MB; Compute parity check per block; Send parity check



FLEX Baseline: BIL/BSQ **to BIP**

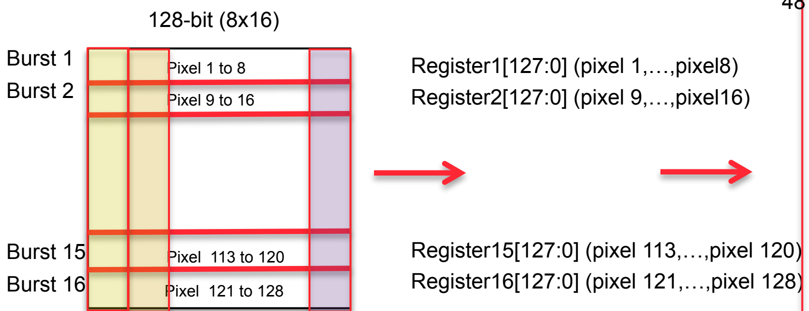
Assumptions:
481 bands
640 samples
32 lines

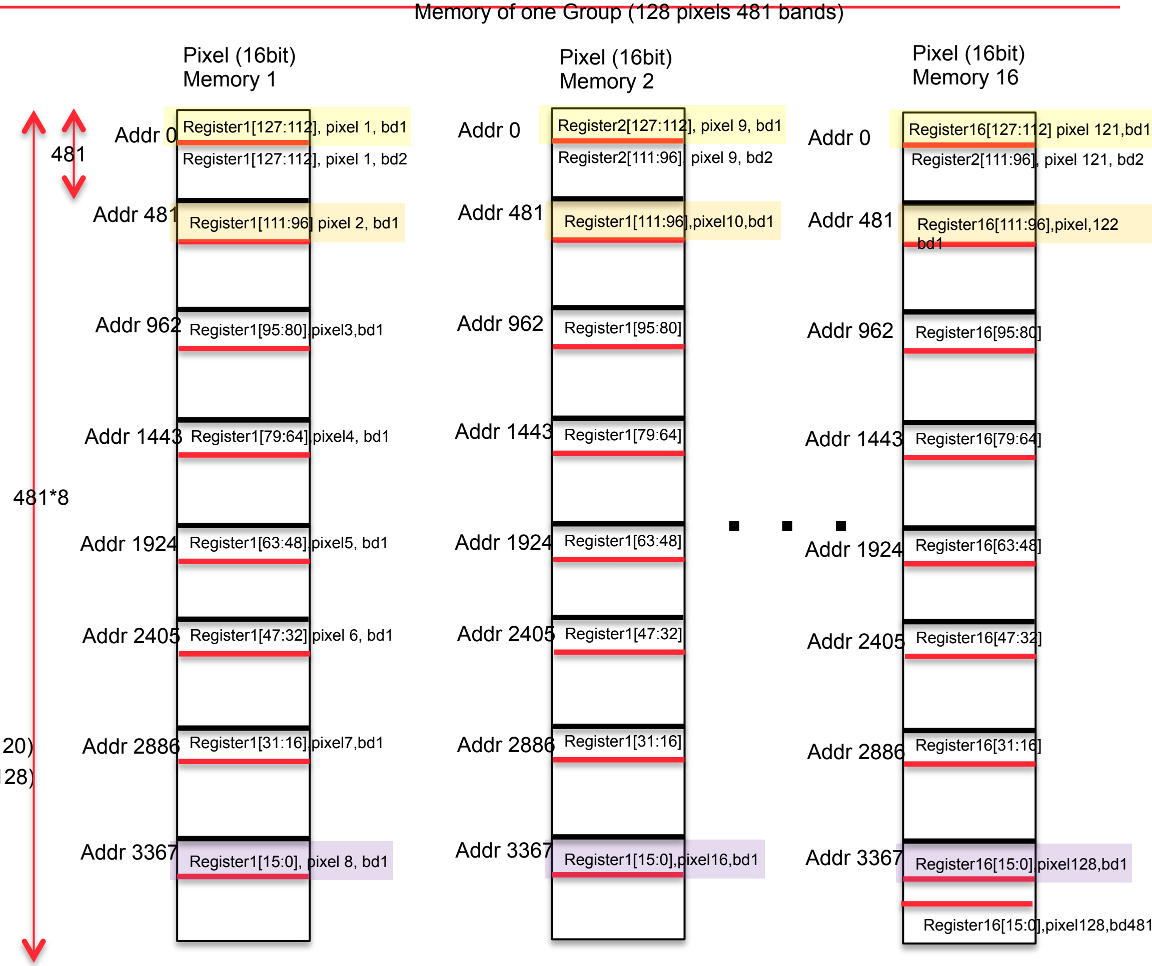


FLEX Baseline: BIL/BSQ to BIP

Read in image
- read in frame-by-frame from DDR memory for 32 frames
- read in group-by-group for 5 groups

Transpose the whole group before going to the next one
- read in burst of 16 128-bit word, saved into 16 registers of width 128 bits
- move data resided in these 16 registers to 16 internal memory of size 3848x16
- total number of bursts to read in one group = 481
  481x(640/5)x16=481x16x128

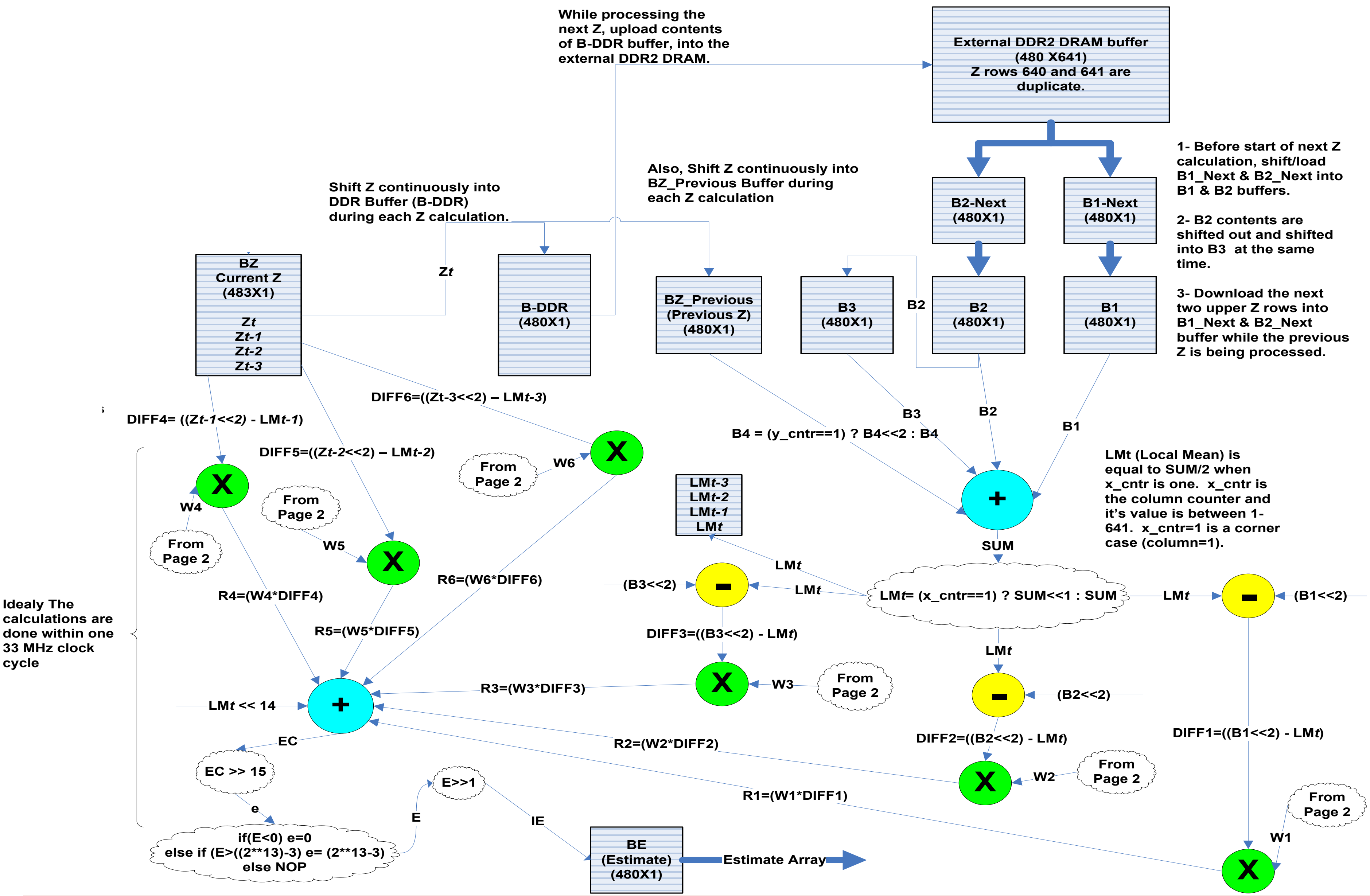




16 internal memory of total size 128pixels X 481bands X 16bits=61568 X16bits=7696 X 128bits
(each memory has data in multiple of 128 bits for Writing to DDR because we chosen 8 pixels)

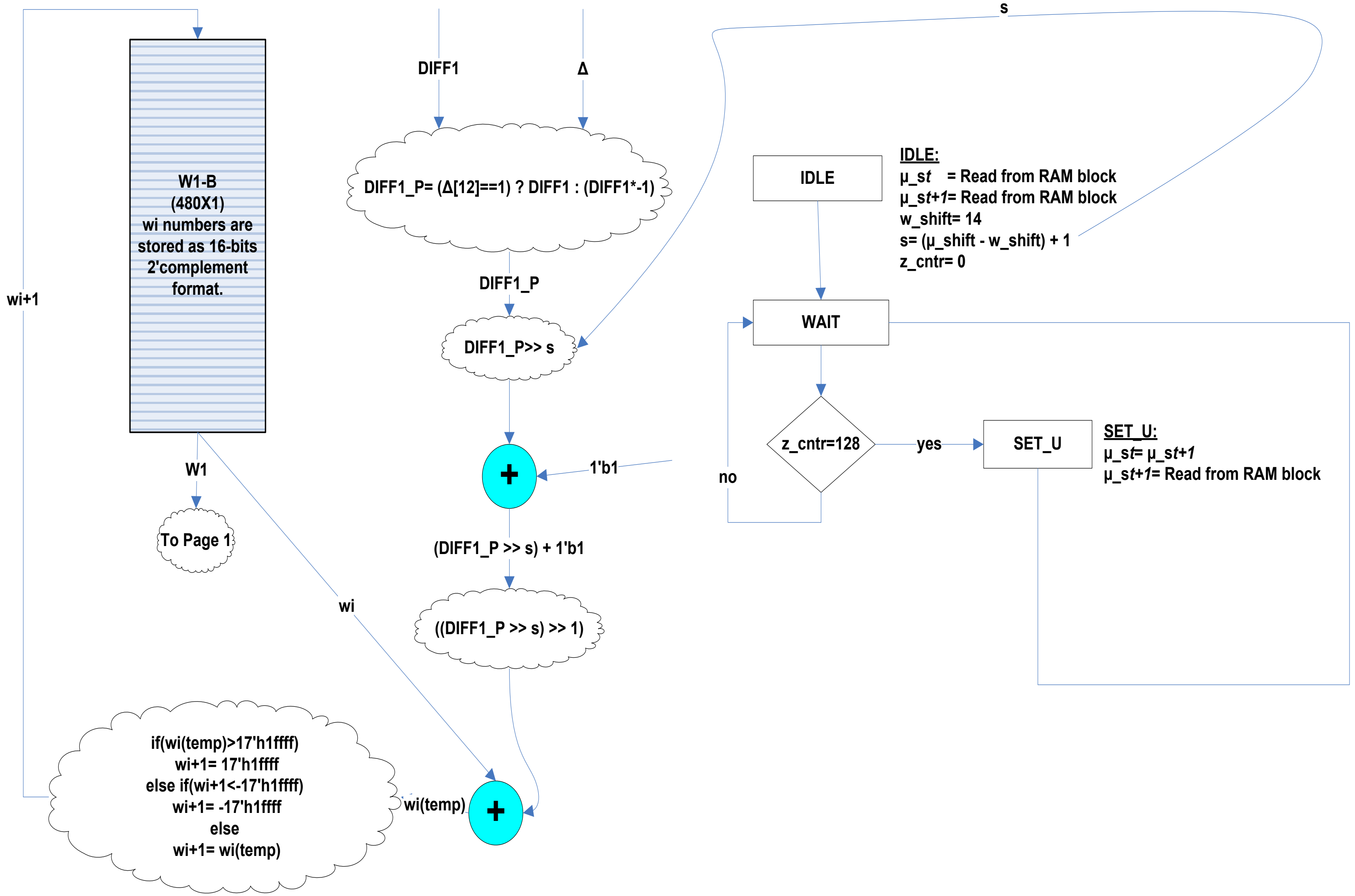
Write out transpose
-write out the whole group transposed to DDR memory
-write out each internal memory at a time
-write out in burst of 64 128-bit words (more efficient with DDR)
-data resided in the internal memory is in multiple of 128-bit
-total number of bursts to move out one Group of internal memory
  = 128pixels X 481bands X 16bits = 7696 X 128 bits
  = 7x64x128 + 1x33x128 (more realistic with burst of 64 in place of 7696)

FLEX Baseline: Predictor IP (from FL)



53

FLEX Baseline: Predictor IP (from FL)



Note: Same logic is used to calculate W2-W6

FL Implementations: FPGA

Demonstration	Instrument				Compressor			
	Name	Frame size	Sampling rate (MS/sec)	Type	Sample Size	FPGA	Throughput * (MS/sec)	Latency* (sec) for 300MB Image
Lab [1, 2]	ARTEMIS			Whiskbroom	12 bits	Virtex4	33	9.09
Airborne [3]	AVIRIS-NG	640X480	30	Pushbroom	13 bits	Virtex5	40	3.75
Airborne [4]	PRISM	640x285	30	Pushbroom	13 bits	Virtex6	40	3.75

ARTEMIS: Advanced Responsive Tactically-Effective Military Imaging Spectrometer

AVIRIS-NG: Airborne Visible/ Infrared Imaging Spectrometer Next Generation

PRISM: Portable Remote Imaging Spectrometer (PRISM) Coastal Ocean Sensor

*Excludes data transfer latency to and from SSD or hard drive

References:

[1] N. Aranki, D. Keymeulen, M. Klimesh, A. Bakhshi, "Fast and Adaptive Lossless On-Board Hyperspectral Data Compression System for Space Applications," *2009 IEEE Aerospace Conference*, 8 pages, March 7-14, 2009, Big Sky, MT, USA.

[2] N. Aranki, D. Keymeulen, M. Klimesh, A. Bakhshi, "Hardware Implementation of Lossless Adaptive and Scalable Hyperspectral Data Compression for Space," *NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 315–322, July, 2009, San Francisco, CA, USA.

[3] A. Bakhshi, J. Kang, N. Aranki, D. Keymeulen, M. Klimesh, A. Kiely "Ecosystem Whitepaper: Implementation of Fast Lossless Hyperspectral Data Compression on Virtex-5 FPGAs", Xilinx on-line January Newsletter 2012.

[4] D. Keymeulen, N. Aranki, A. Bakhshi, H. Luong, C. Sartures, D. Dolman, "Airborne Demonstration of FPGA implementation of Fast Lossless Hyperspectral Data Compression System," *NASA/ESA Conference on Adaptive Hardware and Systems*, July, 2014, Leicester, UK (submitted)

55